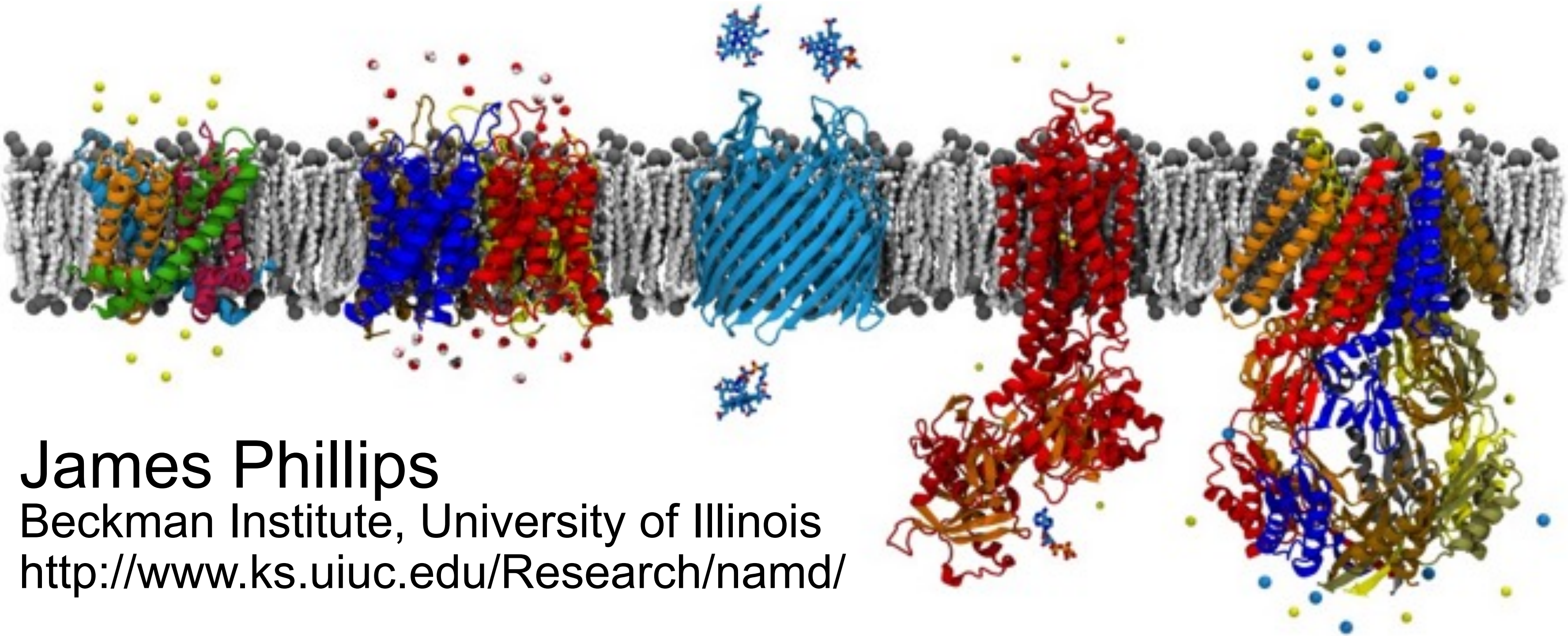
Attacking HIV with Petascale Molecular Dynamics Simulations on Titan and Blue Waters



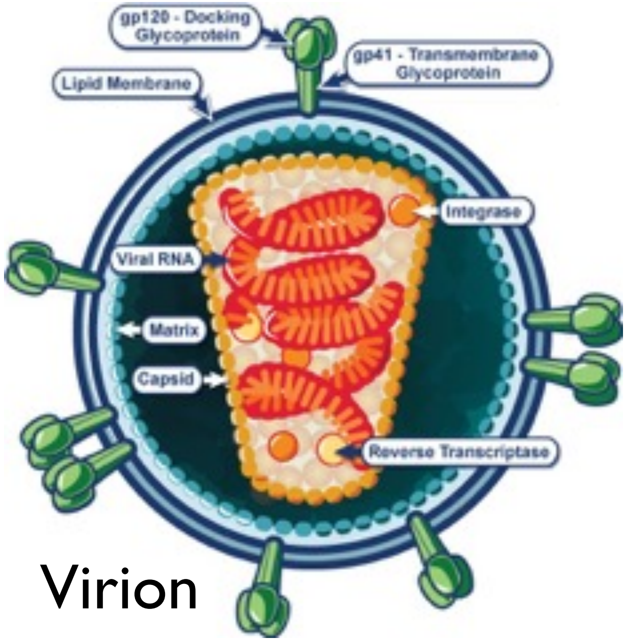
James Phillips

Beckman Institute, University of Illinois

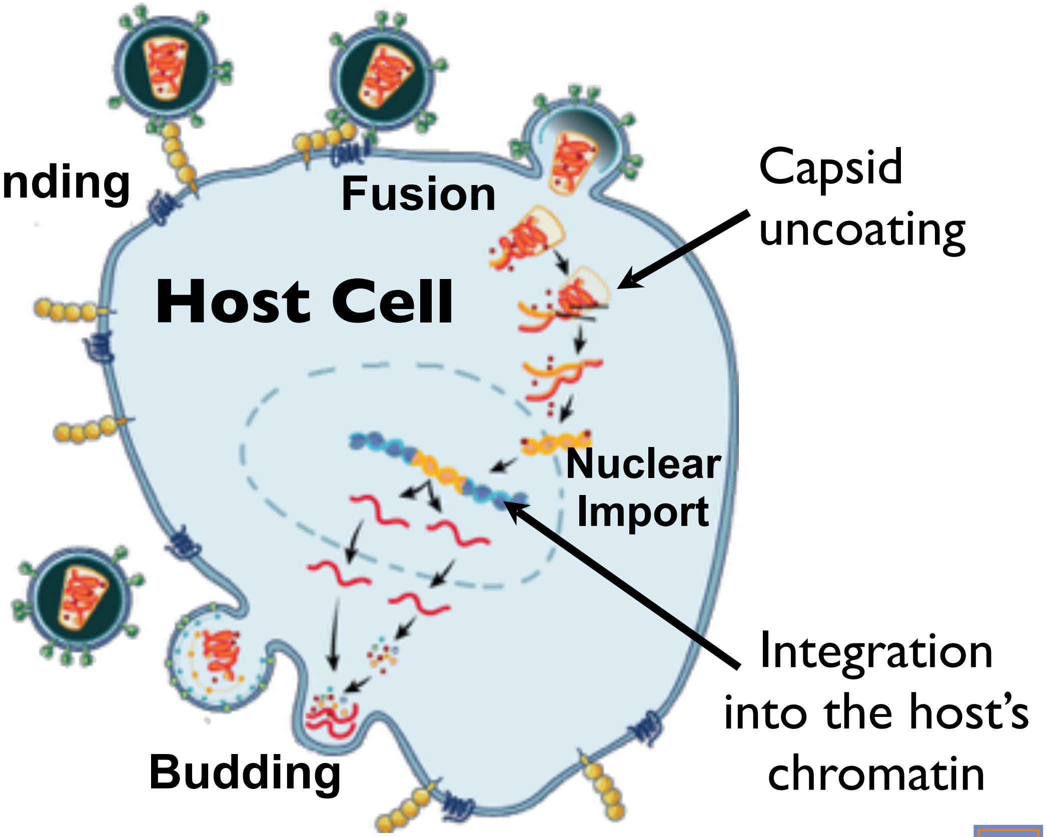
http://www.ks.uiuc.edu/Research/namd/

GTC 2015

HIV Infective Cycle



Virion



Binding

Fusion

Host Cell

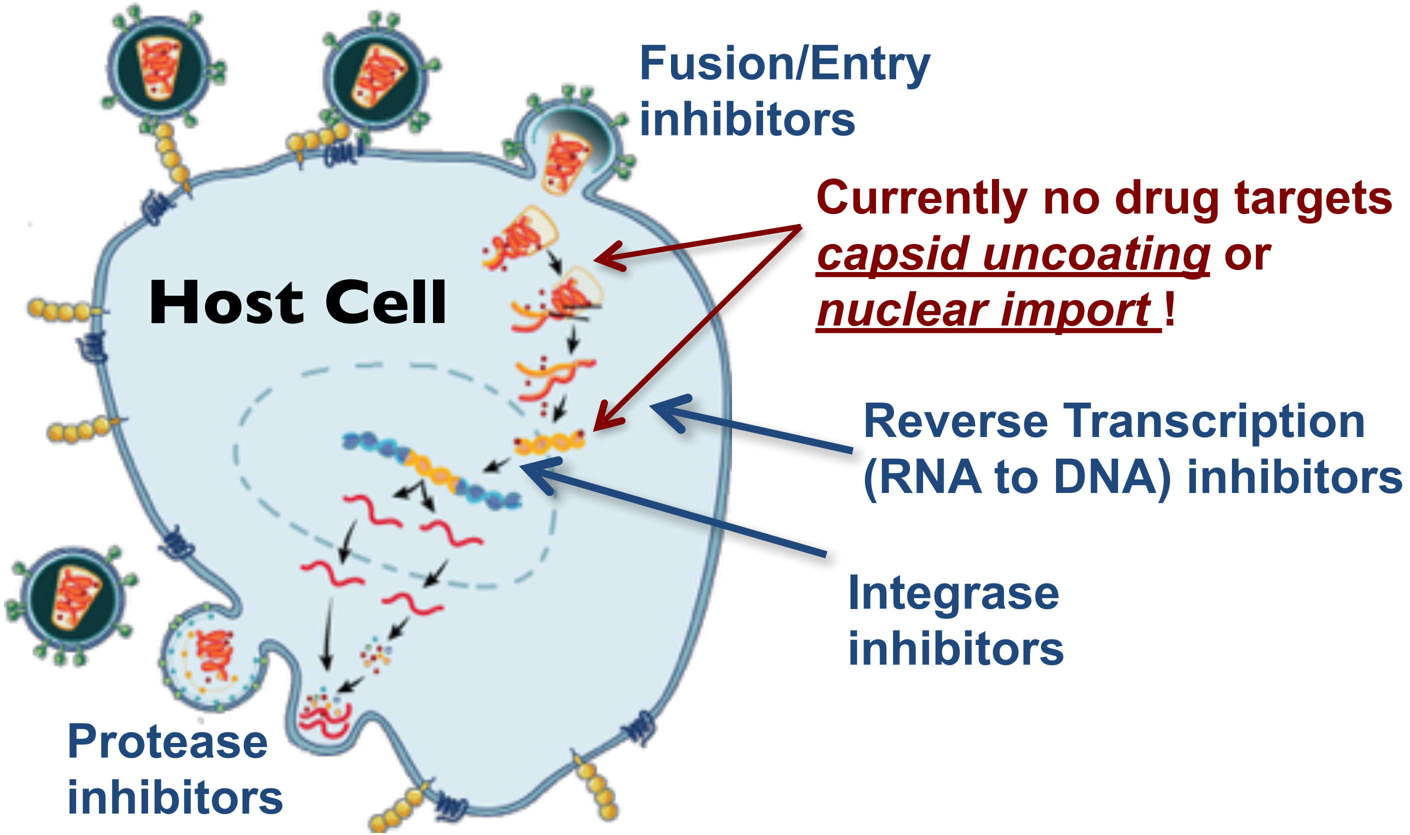
Capsid uncoating

Nuclear Import

Integration into the host's chromatin

Budding

HIV Treatment

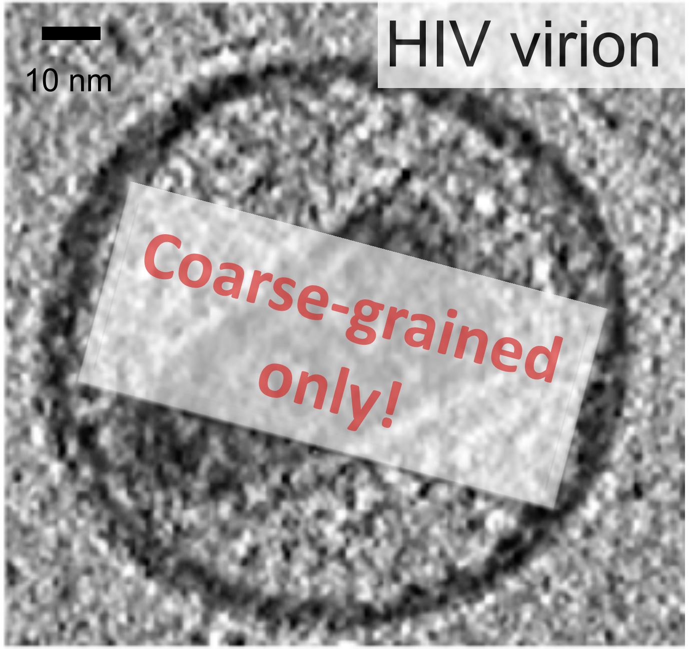

Fusion/Entry inhibitors
Currently no drug targets *capsid uncoating* or *nuclear import* !
Host Cell
Reverse Transcription (RNA to DNA) inhibitors
Integrase inhibitors
Protease inhibitors

HIV Capsid is Much Larger than Previously Simulated Systems

Collaborators:
Peijun Zhang, Angela Gronenborn - U. Pittsburgh
Christopher Aiken - Vanderbilt U.
G. Zhao, et al. *Nature* **497** (2013); exp + comp

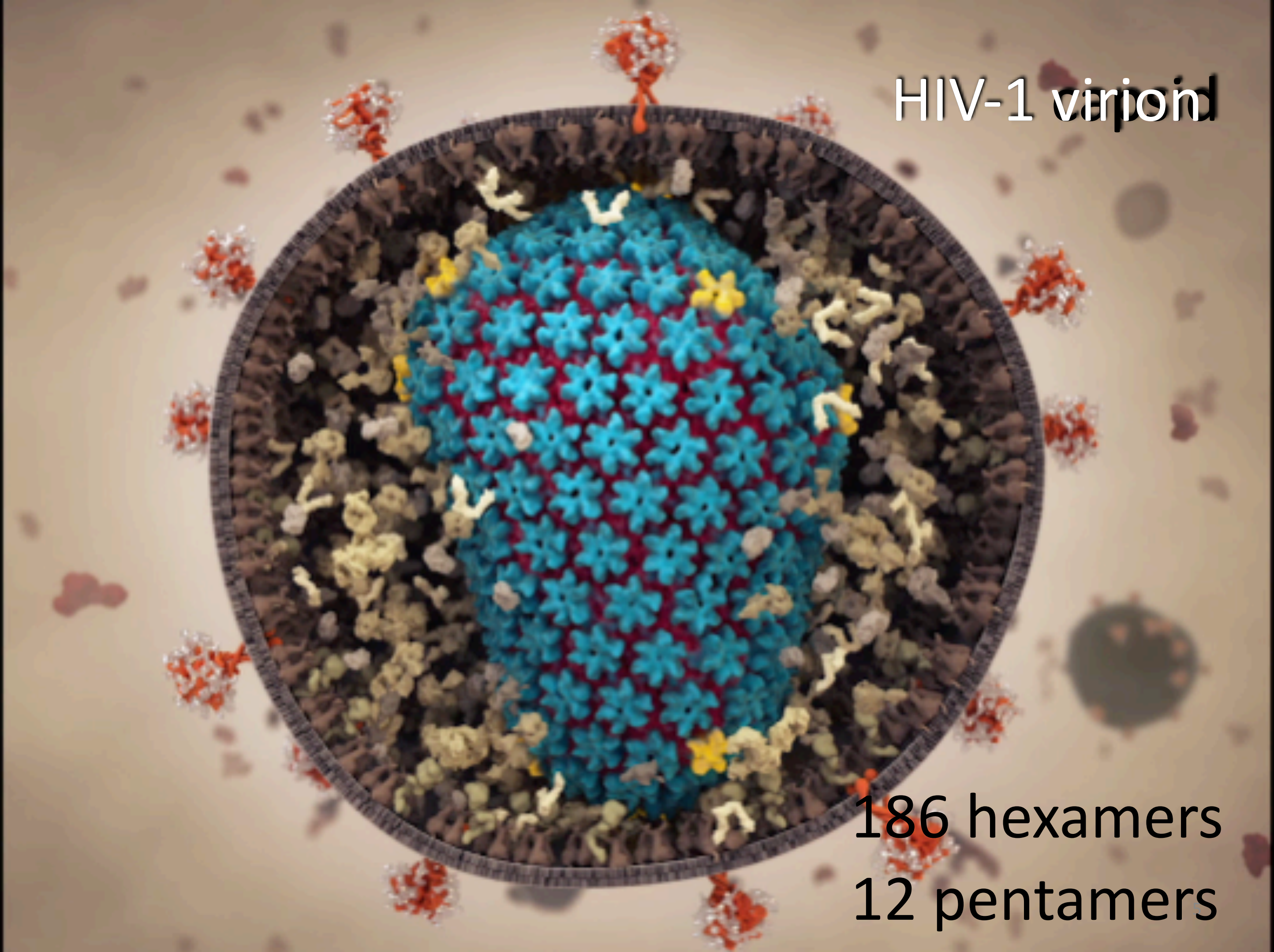
lysozyme

ribosome



All five referees demanded:

Only coarse-grained, not all-atom!



HIV-1 virion

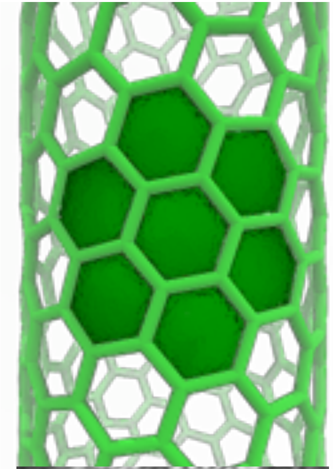
HIV-1 capsid

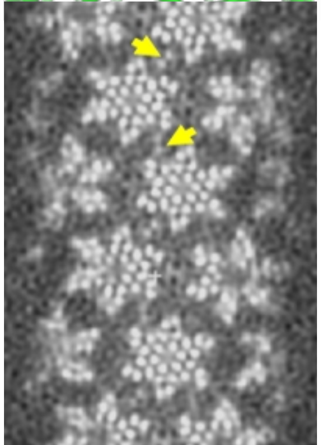
186 hexamers
12 pentamers

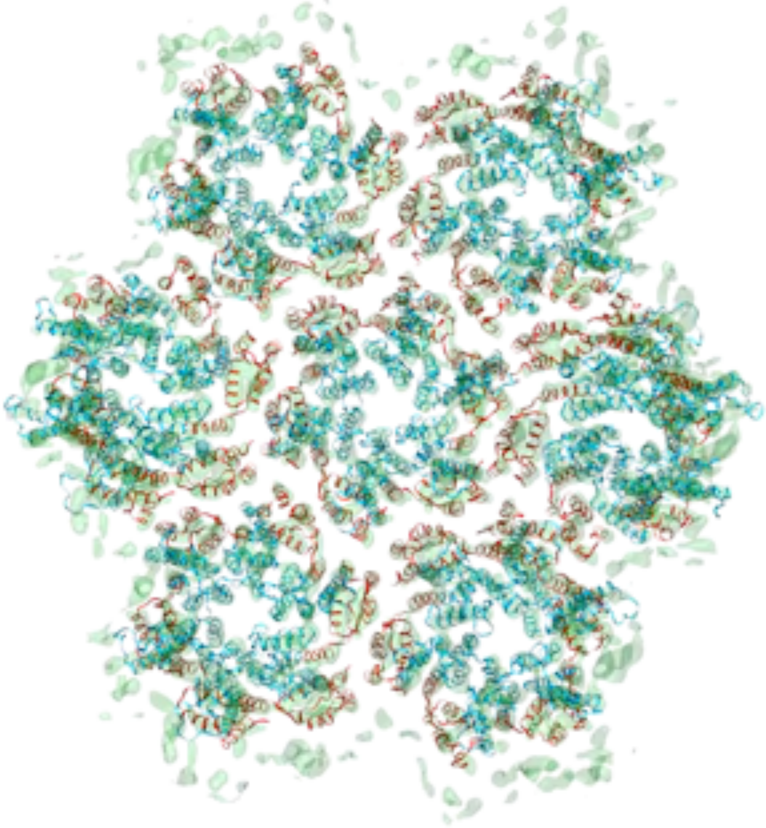
Modeling of the Hexameric Lattice using Molecular Dynamics Flexible Fitting

G. Zhao, et al. *Nature* **497** (2013); exp + comp

Key person:
Juan Perilla
(UIUC)



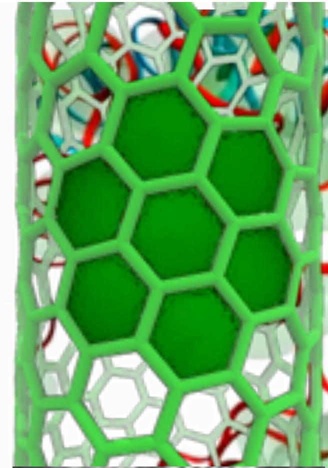


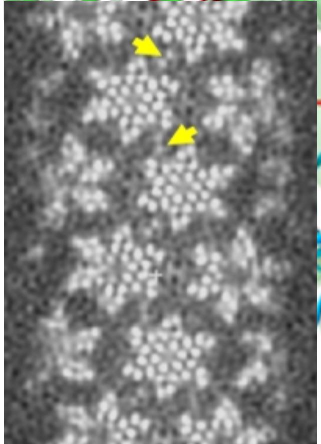


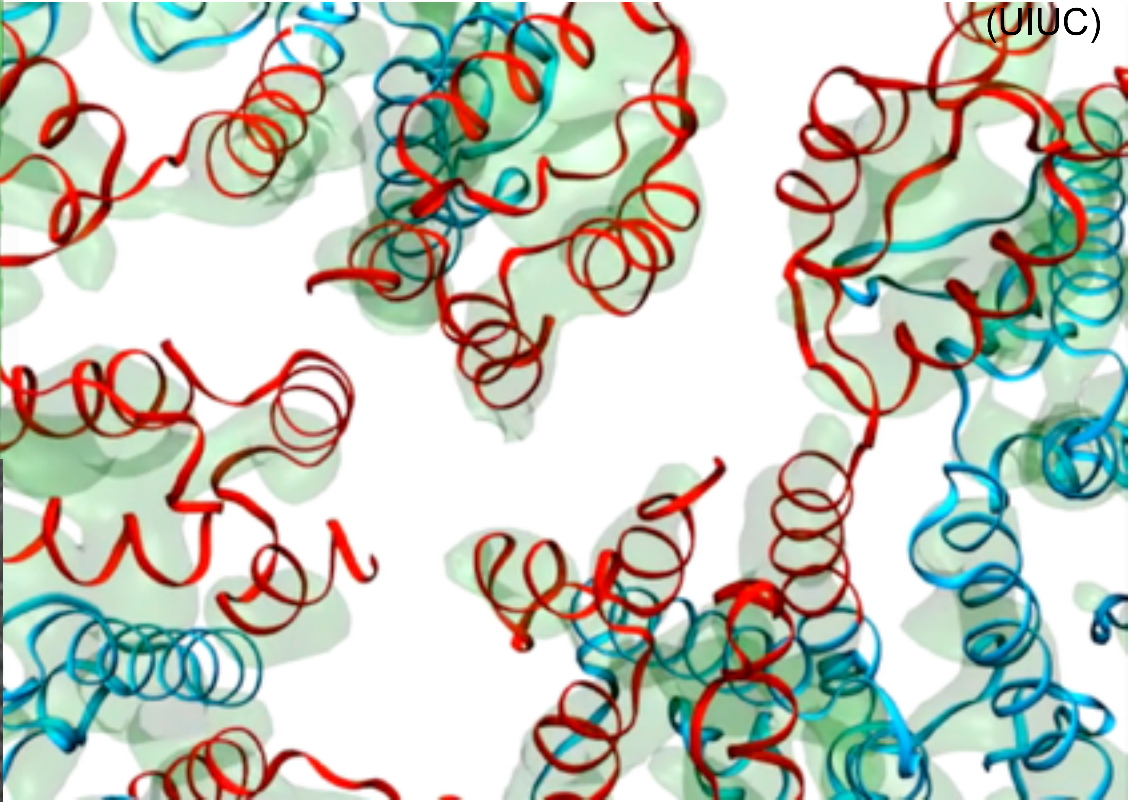
Modeling of the Hexameric Lattice using Molecular Dynamics Flexible Fitting

G. Zhao, et al. *Nature* **497** (2013); exp + comp

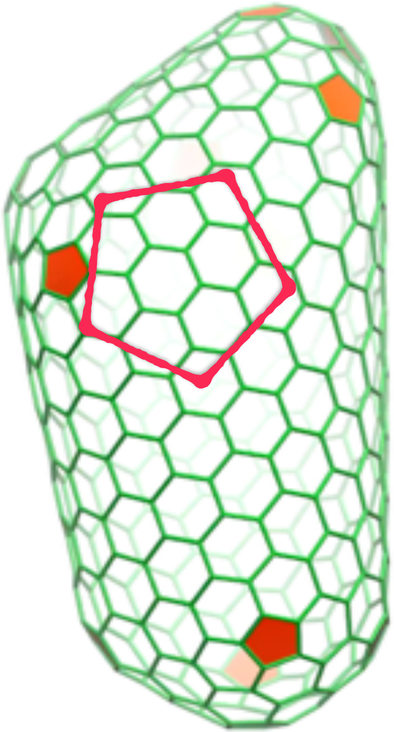
Key person:
Juan Perilla
(UIUC)



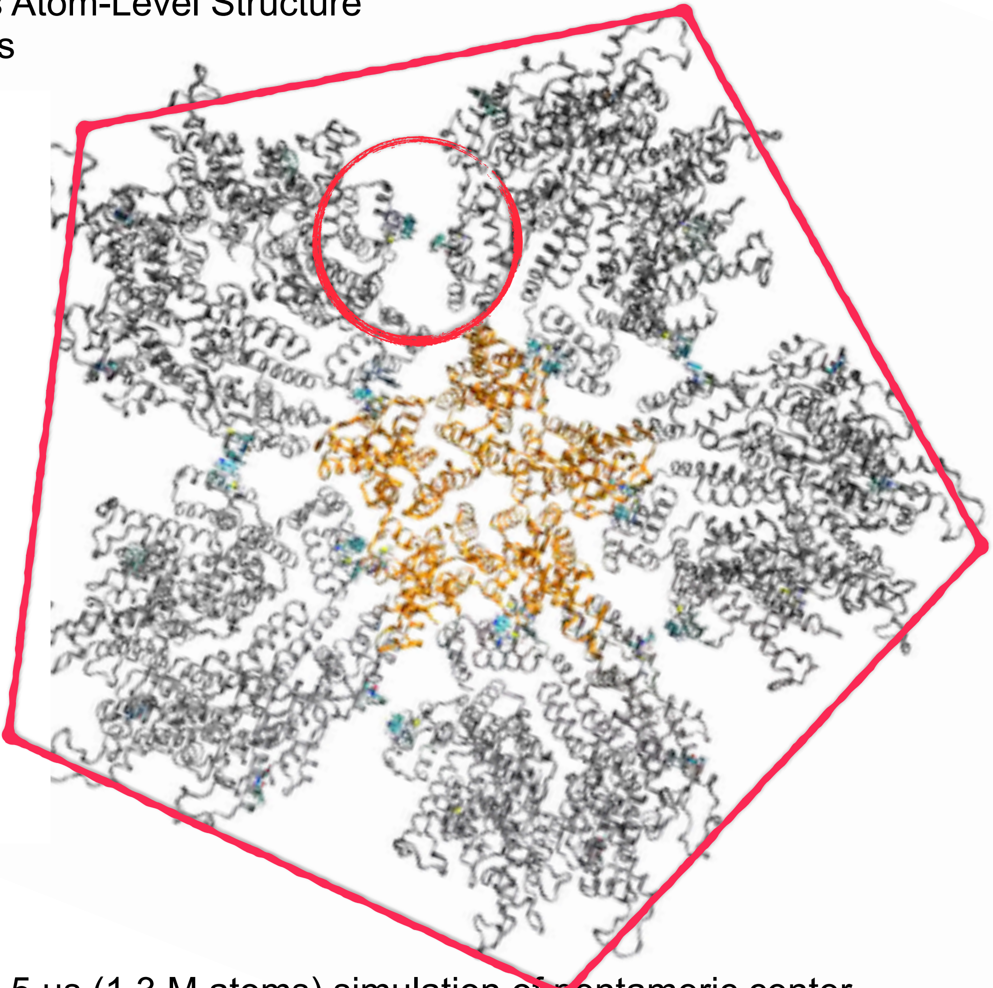




MD Simulation Furnishes Atom-Level Structure of Pentamer-of-Hexamers



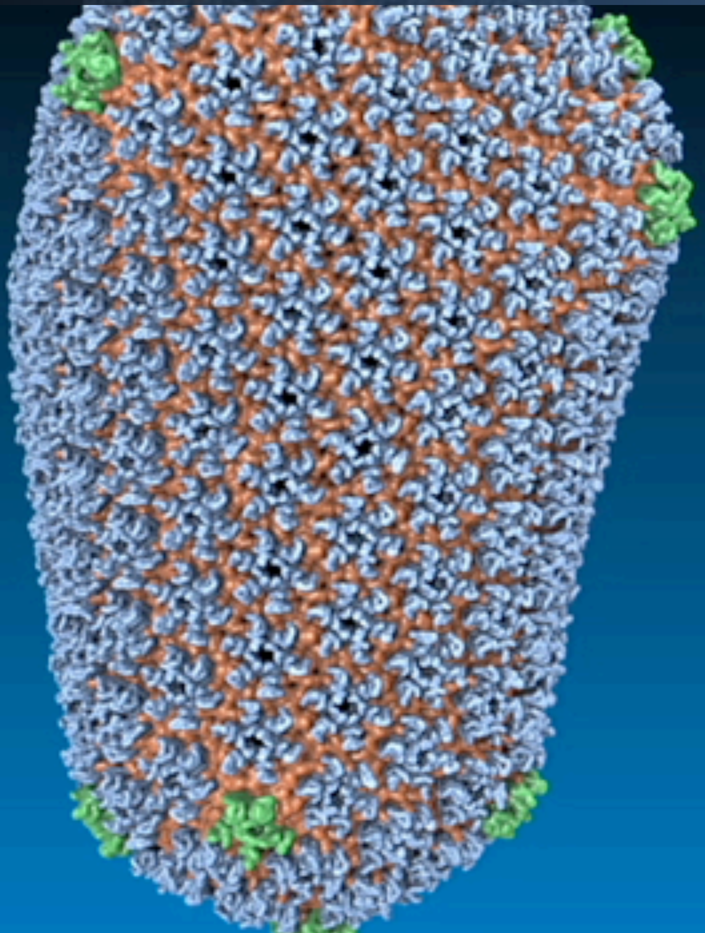
Closed capsid is made of
hexamers-of-hexamers
pentamers-of-hexamers



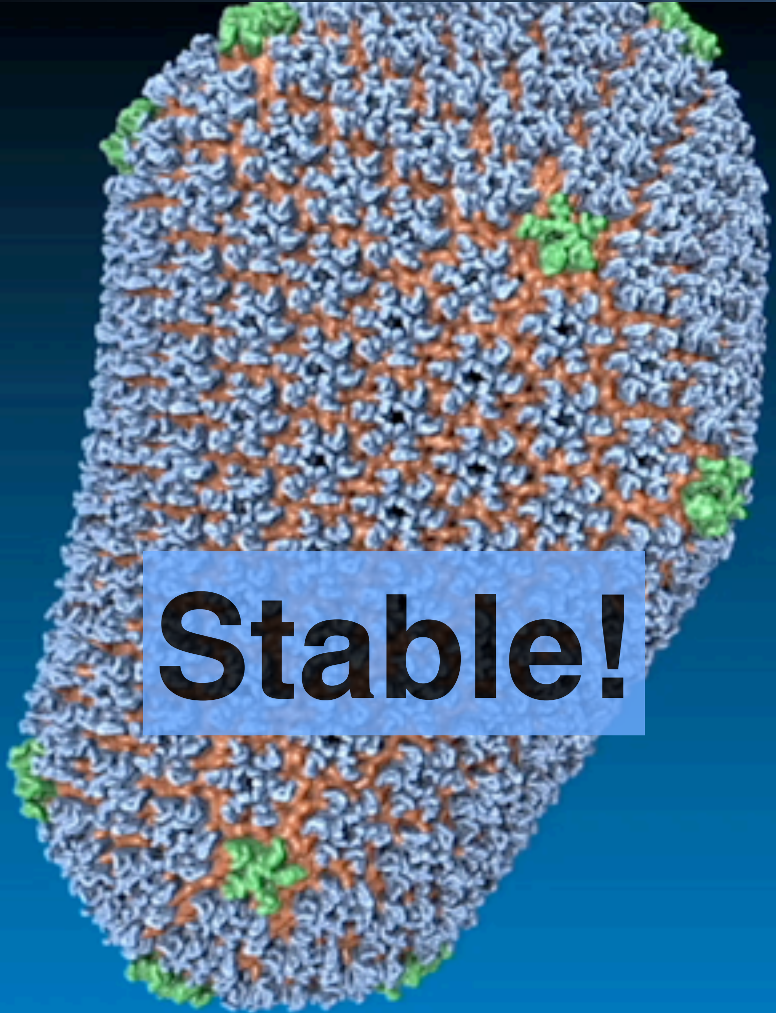
1.5 μs (1.3 M atoms) simulation of pentameric center

HIV capsid contains 186

1300+ proteins,



One-Microsecond Simulation Includes 64 Million Atoms



Stable!

Key person:
Juan Perilla
(UIUC)

nature

THE INTERNATIONAL WEEKLY JOURNAL OF SCIENCE

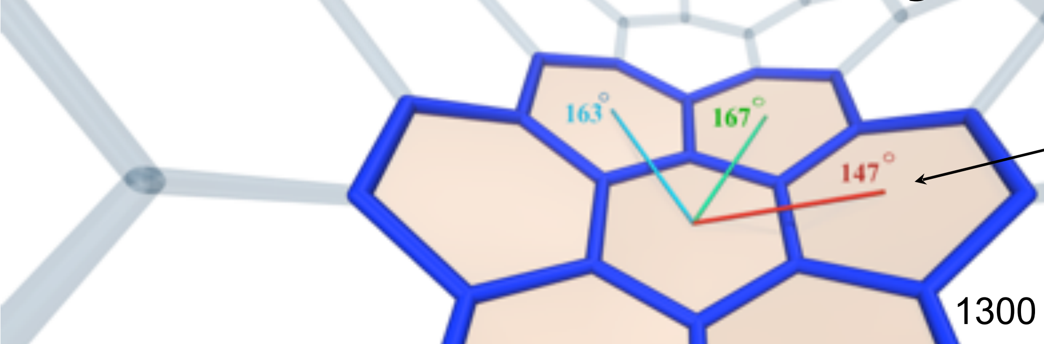
THE HIV-1 CAPSID

Atomic structure of the AIDS pathogen's protein coat

PAGE 643

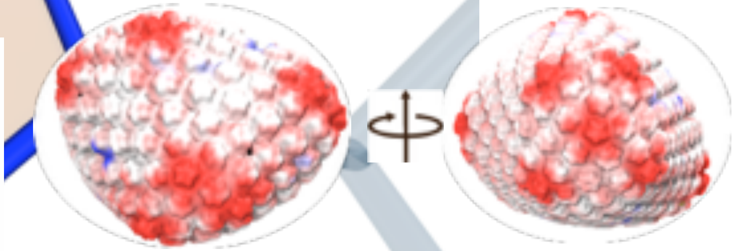
2013 *HPCwire* Editors' Choice Award for Best Use of HPC in Life Sciences

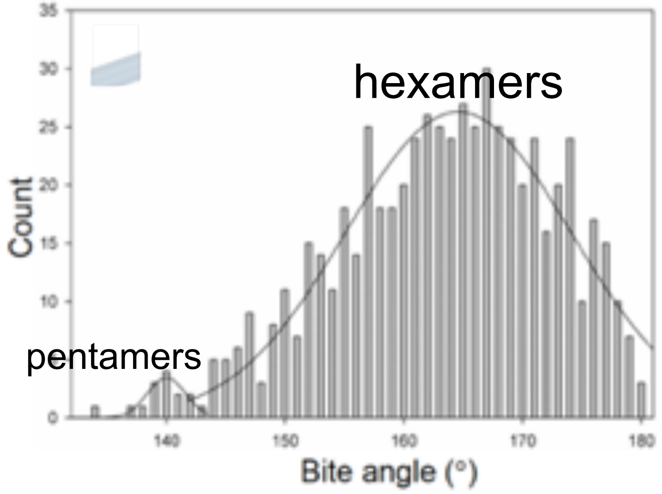
Malleability of HIV-1 CA



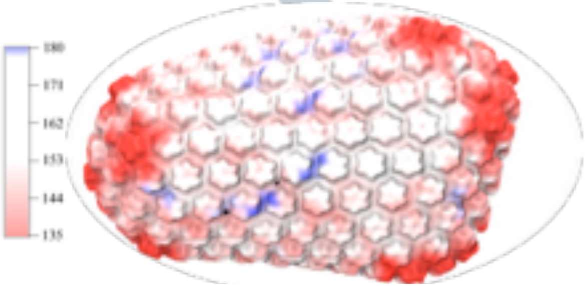
Hexamer of hexamers bite angles along chiral axis

1300 proteins in different conformations



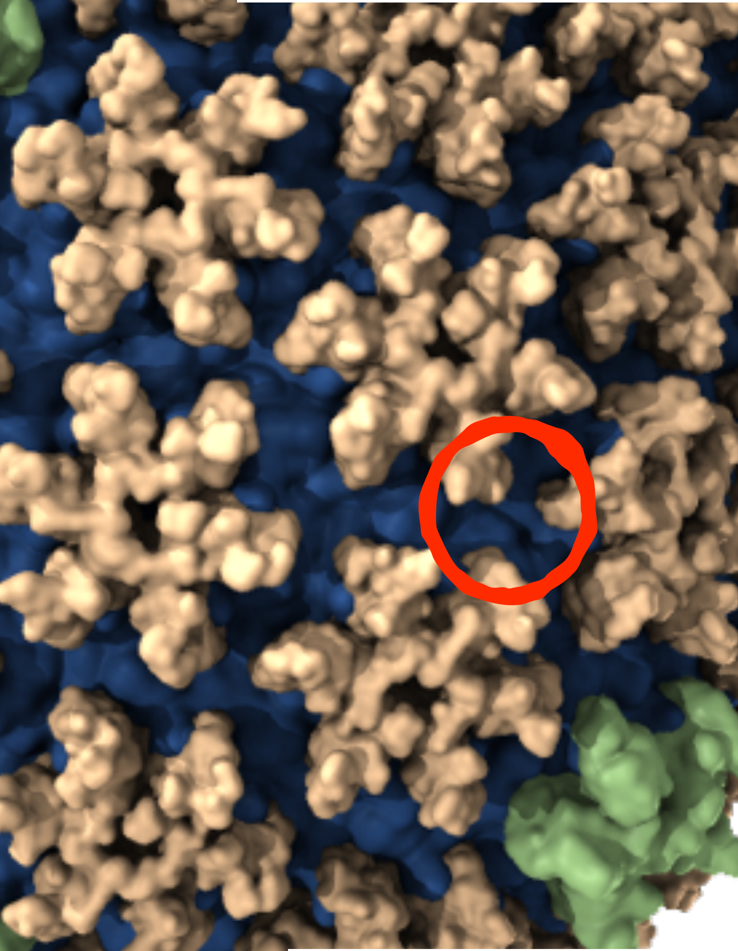


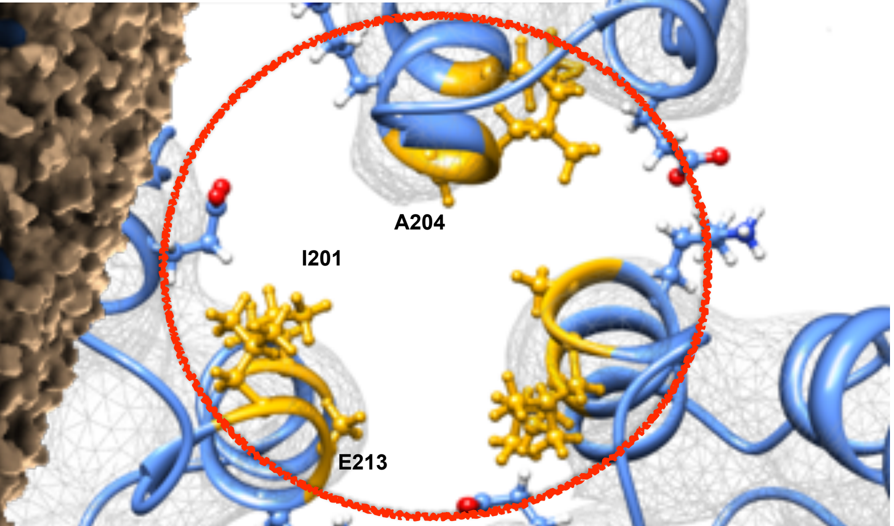
G. Zhao, et al. *Nature* **497** (2013)

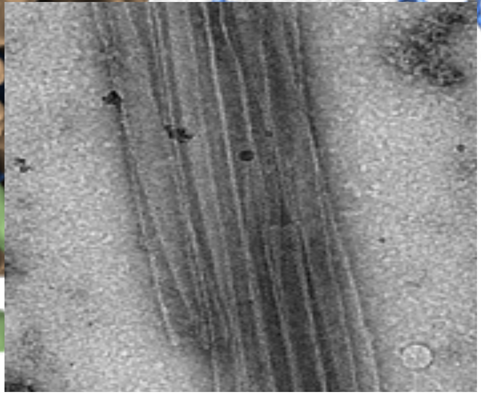


Native capsid bite angle distribution

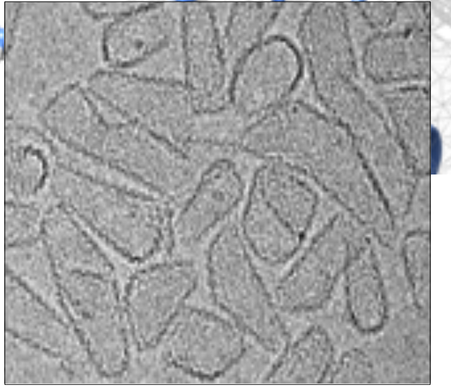
Curvature is regulated by the trimer interface







HIV-CA wild-type *in vitro*

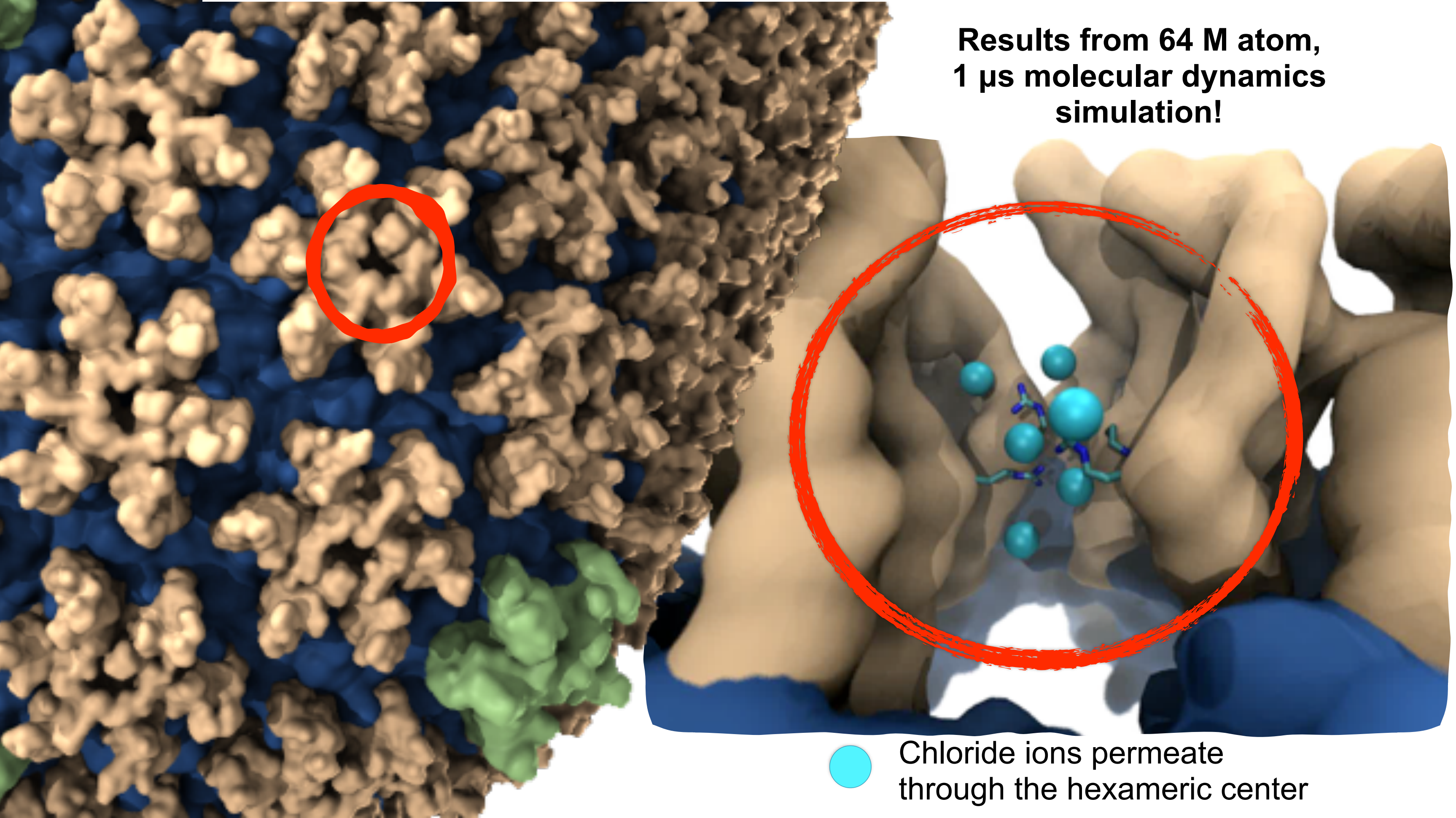


A204C mutant *in vitro*

G. Zhao, et al. *Nature* **497** (2013)

Peijun Zhang - U. Pittsburgh

Capsid acts as an osmotic regulator

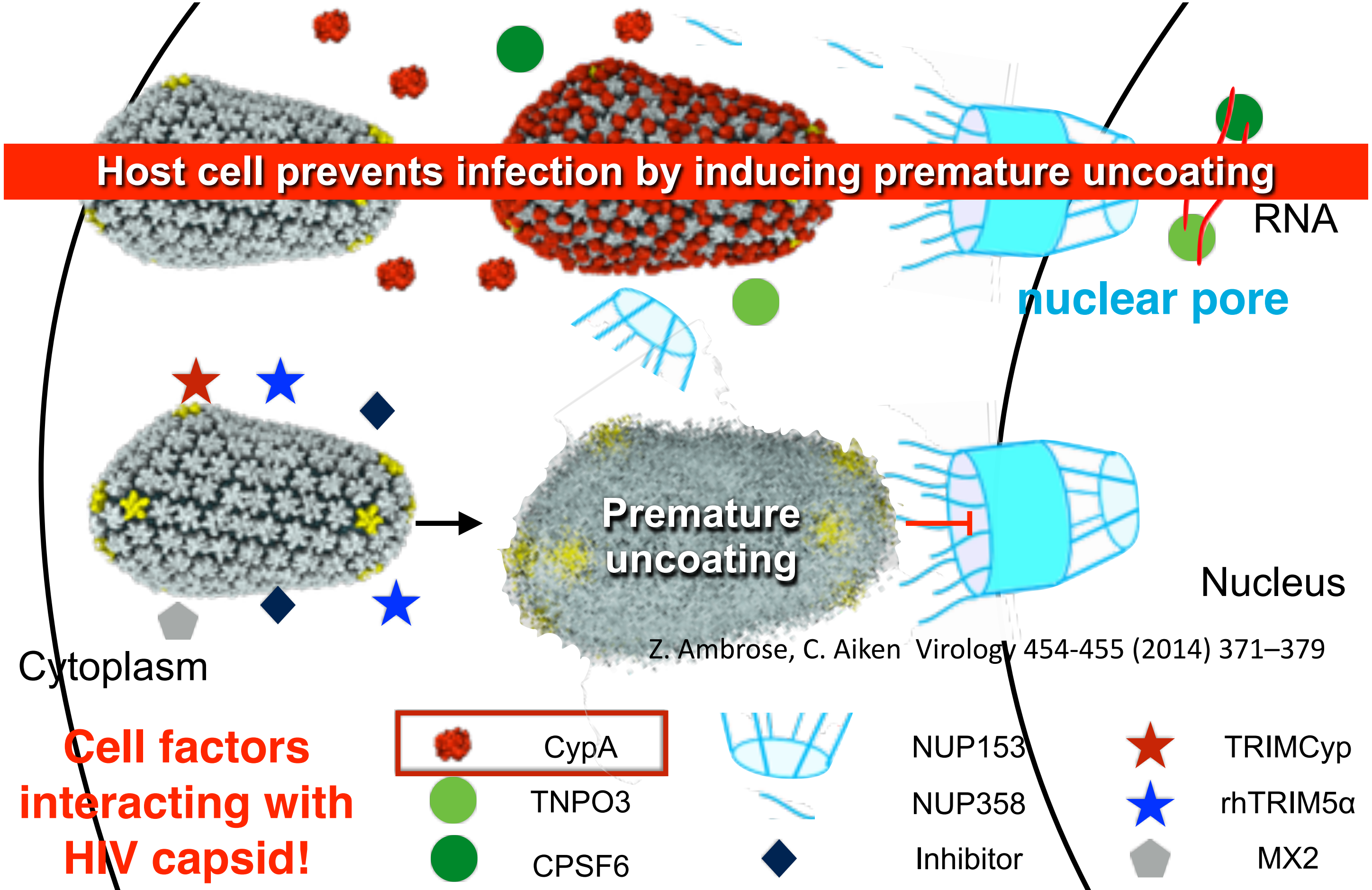


Results from 64 M atom, 1 μs molecular dynamics simulation!

● Chloride ions permeate through the hexameric center

HIV-1 infection

HIV-1 uncoating: regulation by host factors

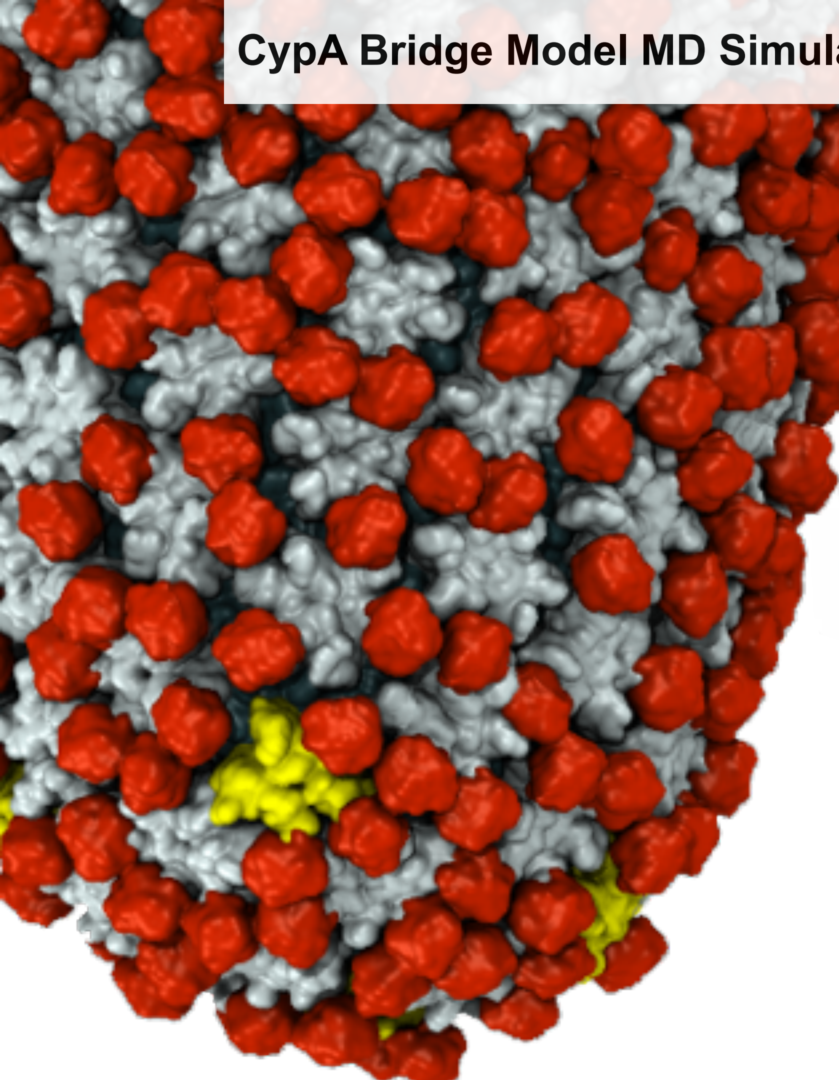


Host cell prevents infection by inducing premature uncoating

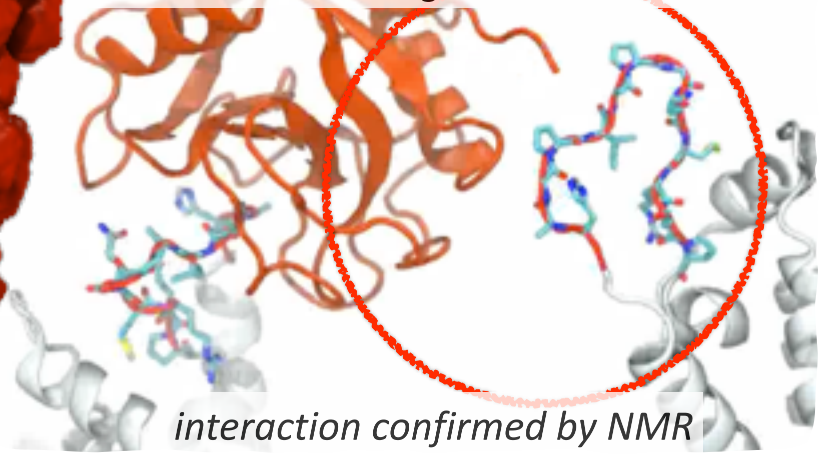
Z. Ambrose, C. Aiken Virology 454-455 (2014) 371–379

Cell factors interacting with HIV capsid!

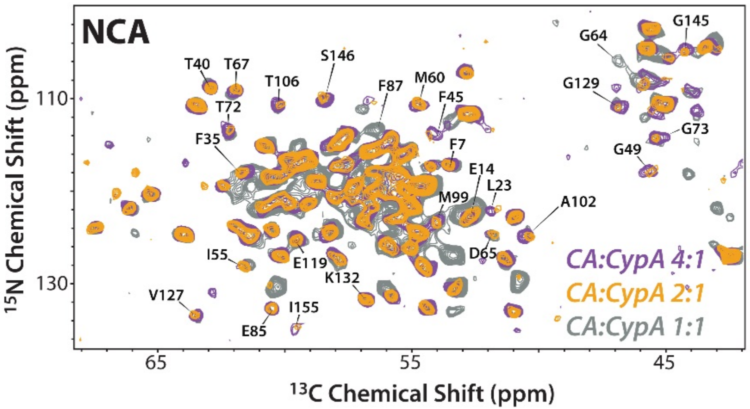
CypA Bridge Model MD Simulations Identify a Novel Catalytic Site



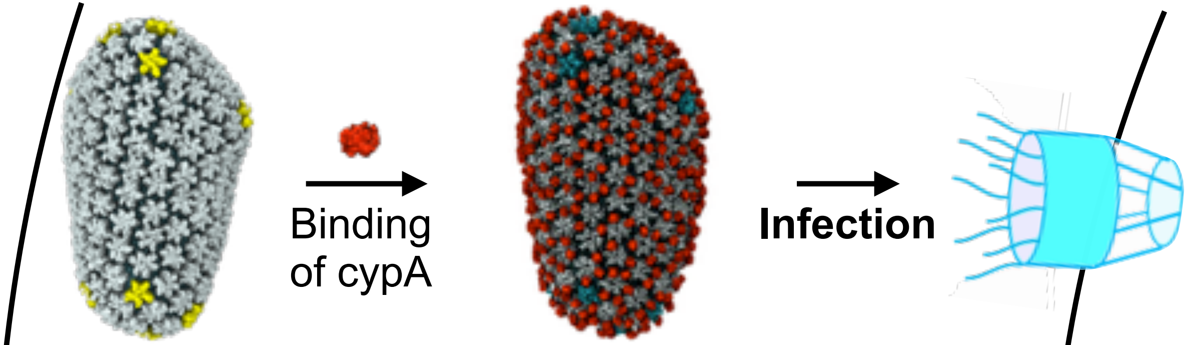
only polarizable force field yields stable bridge interaction



interaction confirmed by NMR



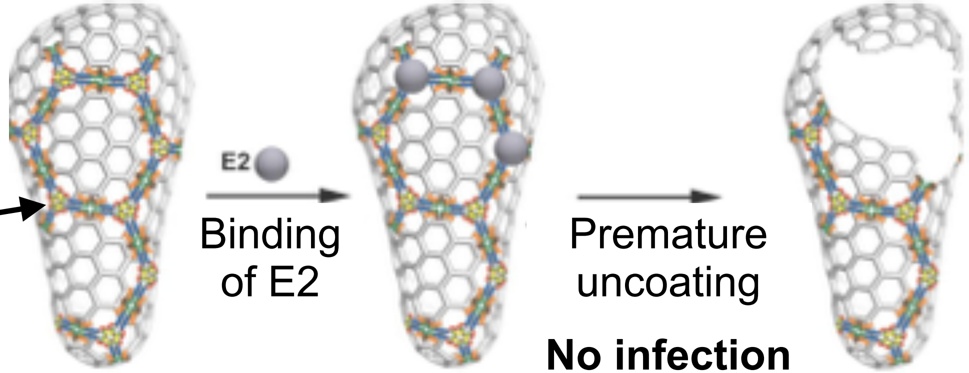
Competitive binding between CypA and TRIM



Binding of cypA

Infection

cypA binding pattern prevents TRIM binding, but leaves Nup interactions intact



TRIM lattice

Binding of E2

Premature uncoating

No infection

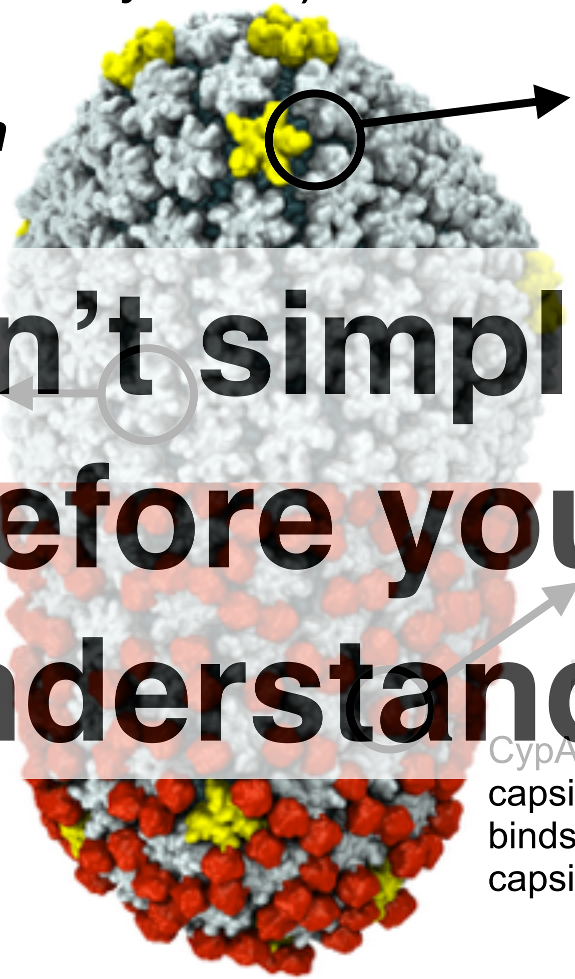
F. Diaz-Griffero, *Viruses* (2011)

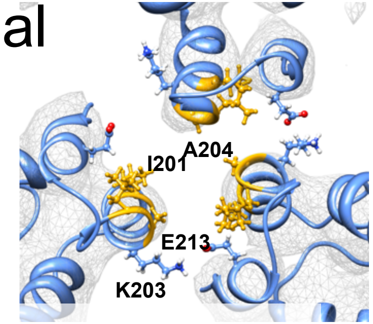
Key person: **Juan Perilla** (UIUC)

Chemical Detail (Every Atom) is Essential for Capsid Role

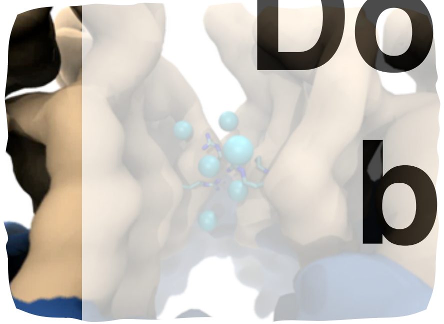
Not always listen to referees!

Don’t simplify before you understand!

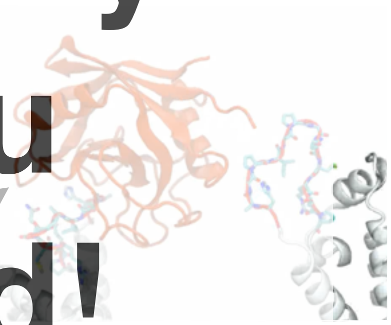




Curvature regulated by trimeric interface



Ions permeate through the capsid



CypA bridges adjacent capsid subunits and thereby binds in particular pattern on capsid surface.

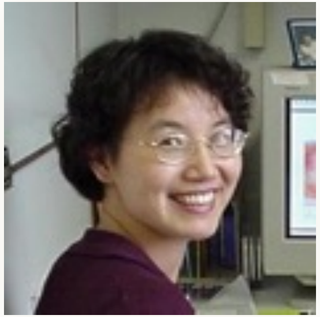
HIV Acknowledgments

Juan R. Perilla **Klaus Schulten**

Theoretical and Computational Biophysics Group

Center for the Physics of Living Cells

University of Illinois at Urbana-Champaign



Peijun Zhang **Angela M. Gronenborn**

Department of Structural Biology
Center for HIV Protein Interactions
University of Pittsburgh School of Medicine

Laxmikant Kale
Parallel Programming Lab
Dept. of Computer Science

Christopher Aiken
Department of Pathology and Immunology
Vanderbilt University School of Medicine

NIH Biomedical Technology Research Center for Macromolecular Modeling and Bioinformatics

Developers of the widely used computational biology software **VMD** and **NAMD**

250,000 registered **VMD** users
72,000 registered **NAMD** users

Renewed 2012-2017 with 10.0 score (NIH)

research projects include: virus capsids, ribosome, photosynthesis, protein folding, membrane reshaping, animal magnetoreception

600 publications (since 1972)
over **54,000** citations

Achievements Built on People

5 faculty members
8 developers
1 systems administrator
17 postdocs
46 graduate students
3 administrative staff

Tajkorshid, Luthey-Schulten, Stone, Schulten, Phillips, Kale, Mallon

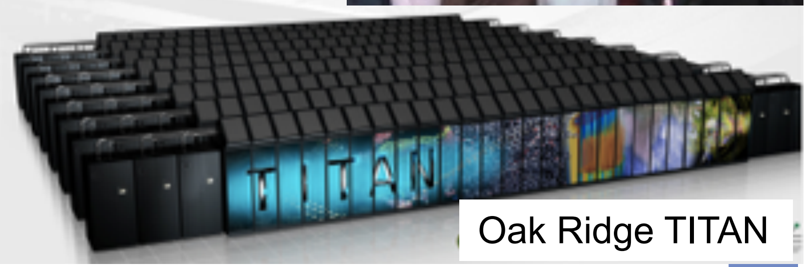
NAMD Serves NIH Users and Goals

Practical Supercomputing for Biomedical Research

- 72,000 users can’t all be computer experts.
  - 18% are NIH-funded; many in other countries.
  - 21,000 have downloaded more than one version.
  - 5000 citations of NAMD reference papers.
- One program available on all platforms.
  - Desktops and laptops – setup and testing
  - Linux clusters – affordable local workhorses
  - Supercomputers – free allocations on XSEDE
  - Blue Waters – sustained petaflop/s performance
  - GPUs - next-generation supercomputing
- User knowledge is preserved across platforms.
  - No change in input or output files.
  - Run any simulation on **any number of cores.**
- Available free of charge to all.

nature
THE INTERNATIONAL WEEKLY JOURNAL OF SCIENCE
THE HIV-1 CAPSID
Atomic structure of the AIDS pathogen's protein coat
PAGE 643

Hands-On Workshops



Oak Ridge TITAN

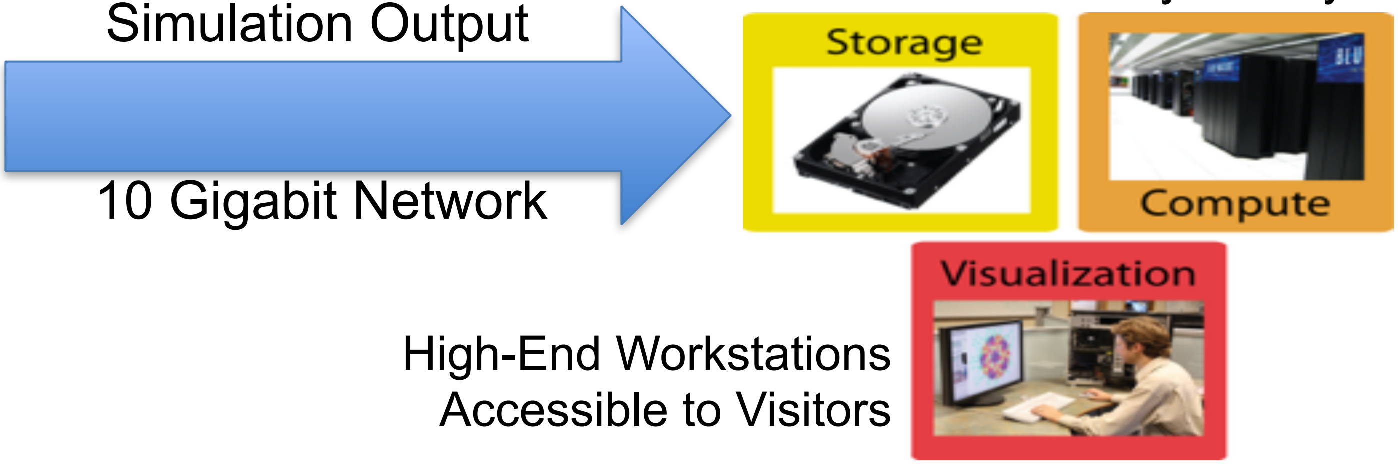
100 Million Atom Simulations Are Not Routine

- Simulation setup is a black art
  - Tools for adding solvent and ions don’t scale
  - Need to move tools and users towards new “js” file format
- Still some rough edges
  - Not all NAMD features usable at scale
- Trajectory and restart output performance
  - New Charm++ I/O library will help address this
- Simulations require leadership machines
  - Available resources are limited, allocation process is slow
- Lack of setup/visualization/analysis facilities

NIH Center Facilities Enable Petascale Biology

Over the past six years the Center has assembled all necessary hardware and infrastructure to prepare and analyze petascale molecular dynamics simulations, and ***makes these facilities available to visiting researchers***.

External Resources,
90% of our Computer
Power



Simulation Output

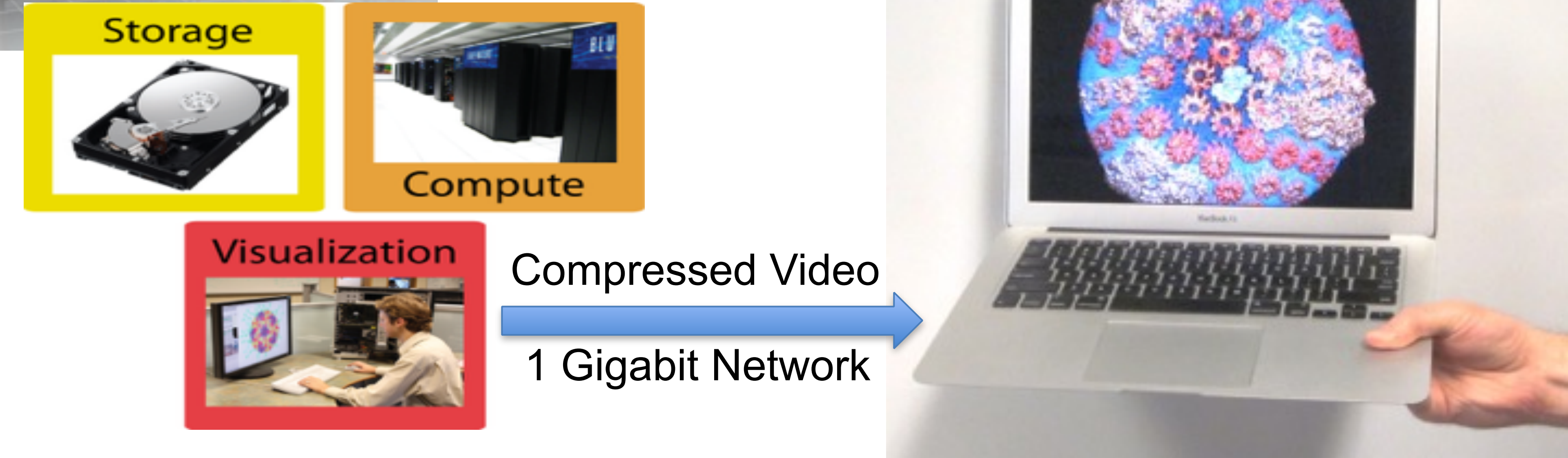
10 Gigabit Network

Petascale Gateway Facility

High-End Workstations
Accessible to Visitors

Virtual Facilities Enable Petascale Anywhere

High-end visualization and analysis workstations currently available only in person at the Beckman Institute must be ***virtualized and embedded at supercomputer centers***.


Storage

Compute

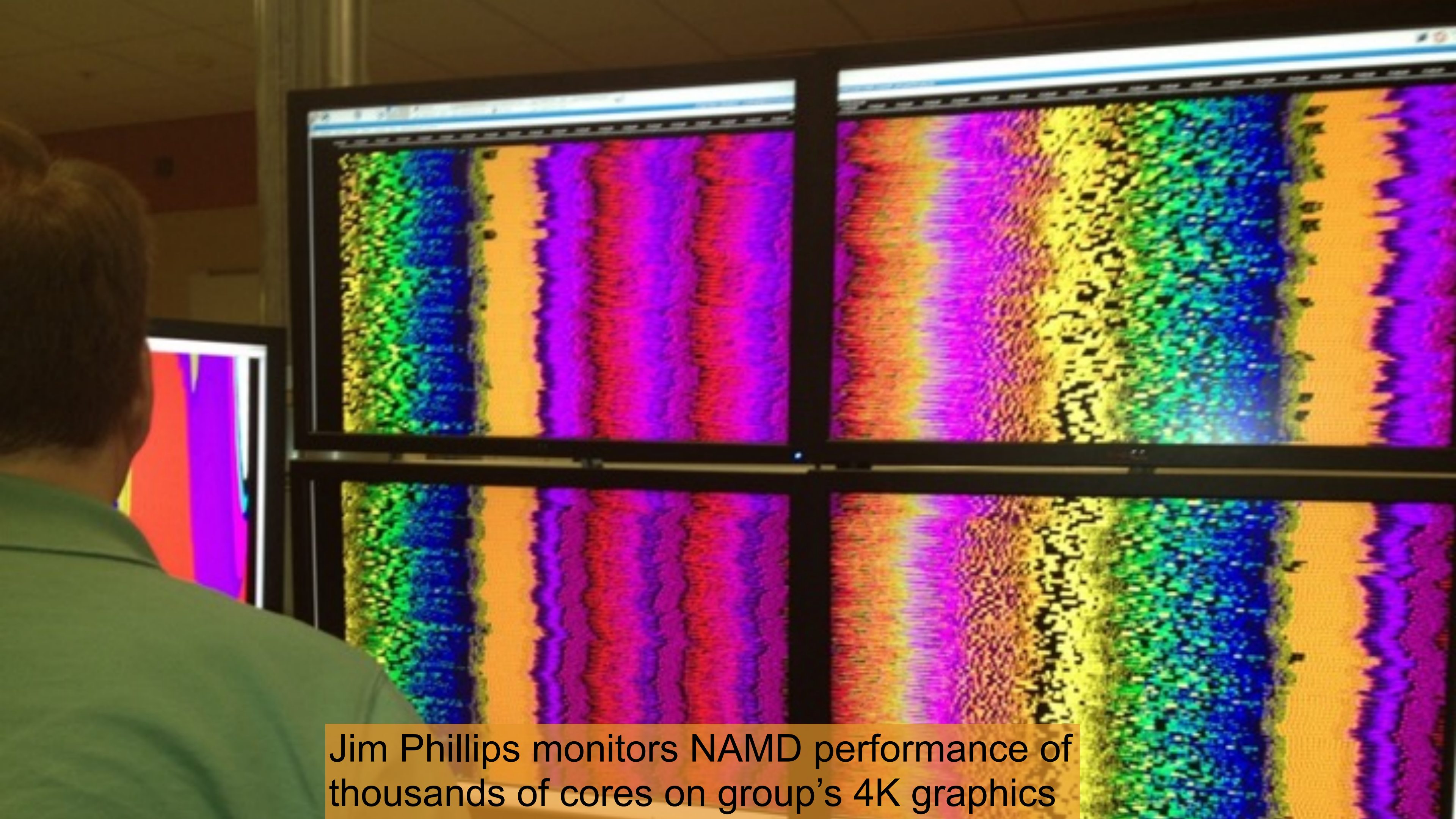
Visualization

Compressed Video

1 Gigabit Network

Remote Visualization Now

- TACC Stampede supports this today
  - Includes nodes with 1TB memory
  - Not virtualized, allocate full dedicated node
  - New Maverick cluster added
- Blue Waters – no visualization resource
- Titan – new Rhea “viz” cluster drops GPUs
- NIH Center - using NICE DCV for remote access

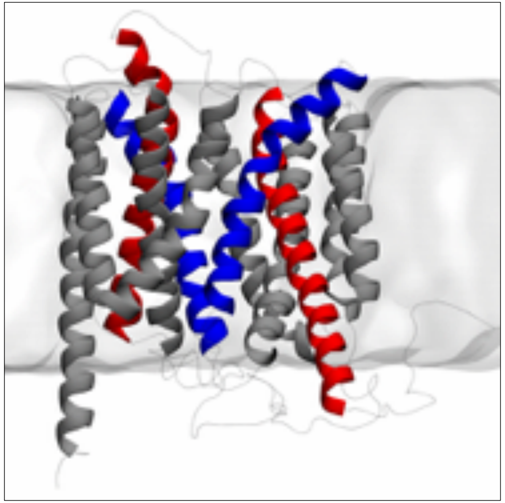


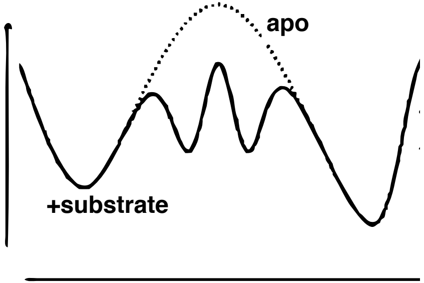
Jim Phillips monitors NAMD performance of thousands of cores on group’s 4K graphics

NAMD 2.10 Release (December 2014)

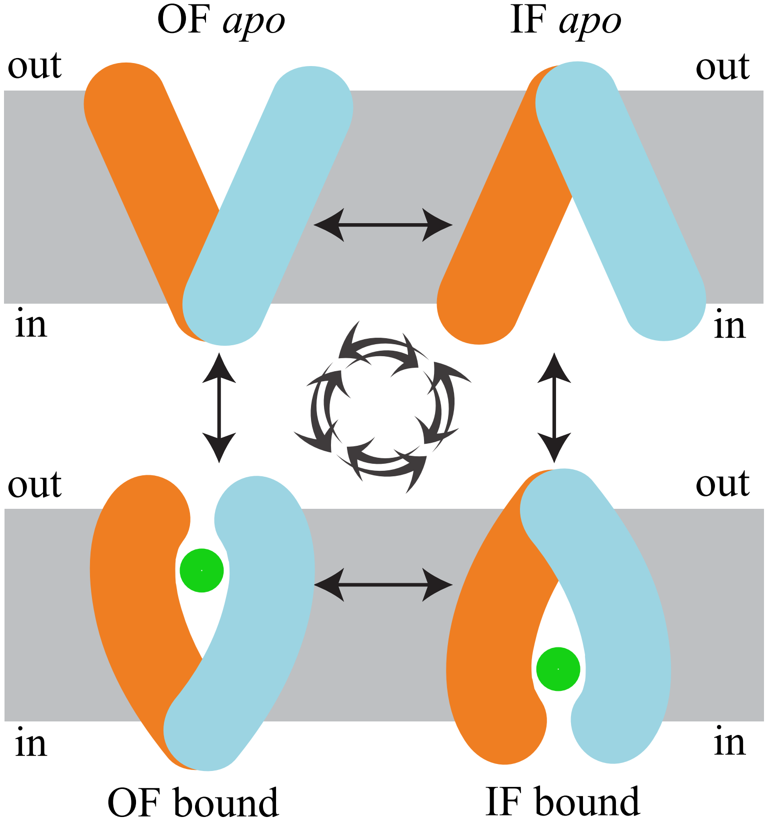
- Focus on enabling petascale simulations
- Type 1: Large systems of ~100 million atoms
  - Scalable to all of Blue Waters or Titan (Phillips et al., SC14)
  - **In regular production use for multiple biomedical driving projects**
  - Amaro (UCSD) allocation on Blue Waters for 210M-atom influenza virus
- Type 2: Replica exchange simulations of smaller systems
  - Improved performance over NAMD 2.9, especially with GPUs
  - Scalable multiple copy algorithms, *Comp. Phys. Comm.* 185:908-16
  - Multiple-walker adaptive biasing force, *J. Chem. Theo. Comp.* 10:5276-85
  - Adaptive multilevel splitting, *ESAIM Proc.* (in press)
- Various other improvements
  - Xeon Phi port, GPU improvements including PME offload
  - Semi- and non-periodic long-range electrostatics (multilevel summation)

NAMD Replica Exchange Example Application: Complete Description of Transport Cycle

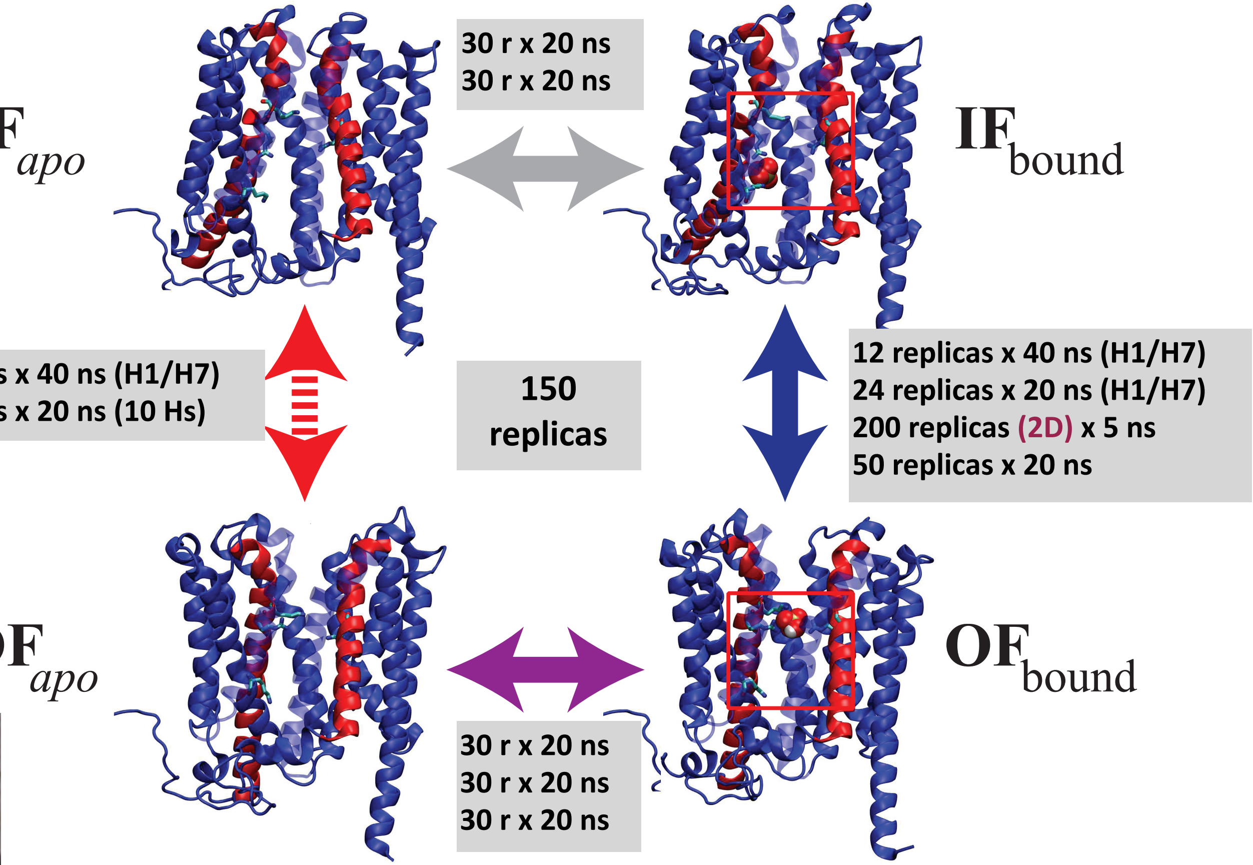




Law, *et al.*, *Biochemistry* **46**, 12190 (2007).



Advanced Replica Exchange Simulation Protocol
Requiring a Combination of Multiple Collective Variables



IF$_{apo}$

30 r x 20 ns
30 r x 20 ns

IF$_{bound}$

12 replicas x 40 ns (H1/H7)
50 replicas x 20 ns (10 Hs)

150
replicas

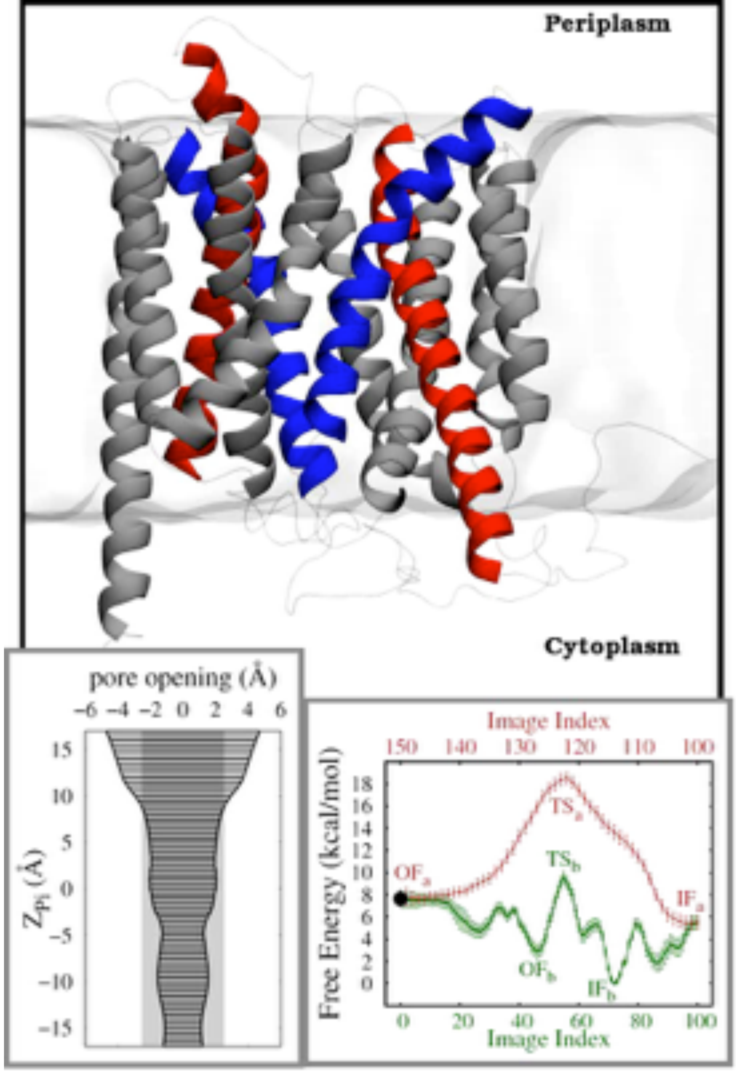
12 replicas x 40 ns (H1/H7)
24 replicas x 20 ns (H1/H7)
200 replicas (2D) x 5 ns
50 replicas x 20 ns

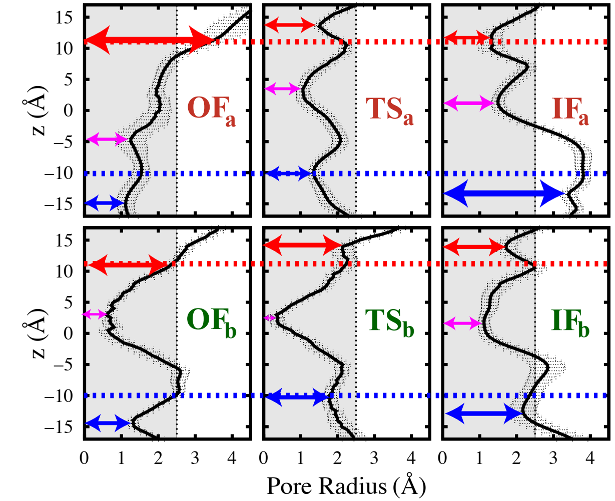
OF$_{apo}$

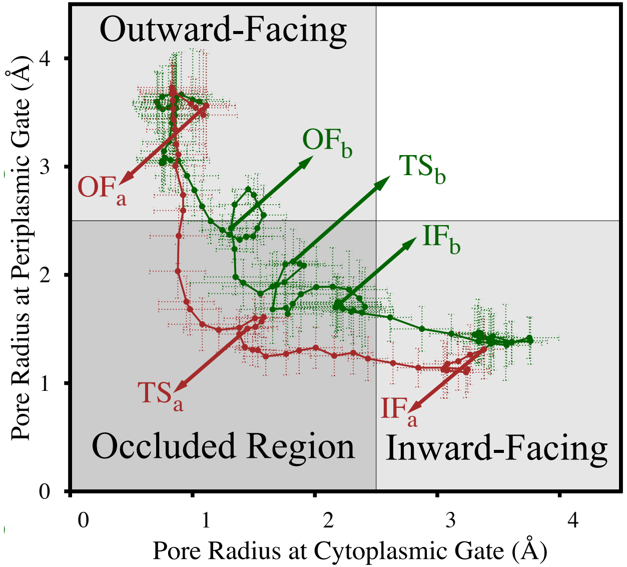
OF$_{bound}$

30 r x 20 ns
30 r x 20 ns
30 r x 20 ns

Mahmoud Moradi


Periplasm
Cytoplasm
pore opening (Å)
−6 −4 −2 0 2 4 6
15
10
5
0
−5
−10
−15
Z_{Pi} (Å)
Image Index
150 140 130 120 110 100
Free Energy (kcal/mol)
18 16 14 12 10 8 6 4 2 0
OF_a
TS_a
IF_a
TS_b
OF_b
IF_b
0 20 40 60 80 100
Image Index


15
10
5
0
−5
−10
−15
z (Å)
OF_a
TS_a
IF_a
OF_b
TS_b
IF_b
0 1 2 3 4
Pore Radius (Å)


Outward-Facing
Occluded Region
Inward-Facing
OF_a
TS_a
IF_a
OF_b
TS_b
IF_b
Pore Radius at Periplasmic Gate (Å)
0 1 2 3 4
Pore Radius at Cytoplasmic Gate (Å)
0 1 2 3 4

Computational Structural Biology and Molecular Biophysics Group (CSBMB)

csbmb.beckman.illinois.edu

Mahmud Moradi
Giray Enkavi
Jing Li
Po-Chao Wen
Sundar Thangapandian
Noah Trebesch

Collaborating Labs
H. Mchaourab (Vanderbilt)
R. Nakamoto (U. Virginia)
D.-N. Wang (NYU)

XSEDE
Extreme Science and Engineering Discovery Environment

BLUE WATERS NCSA

NATIONAL INSTITUTES OF HEALTH

mpsdc

U.S. DEPARTMENT OF ENERGY
INCITE
LEADERSHIP COMPUTING

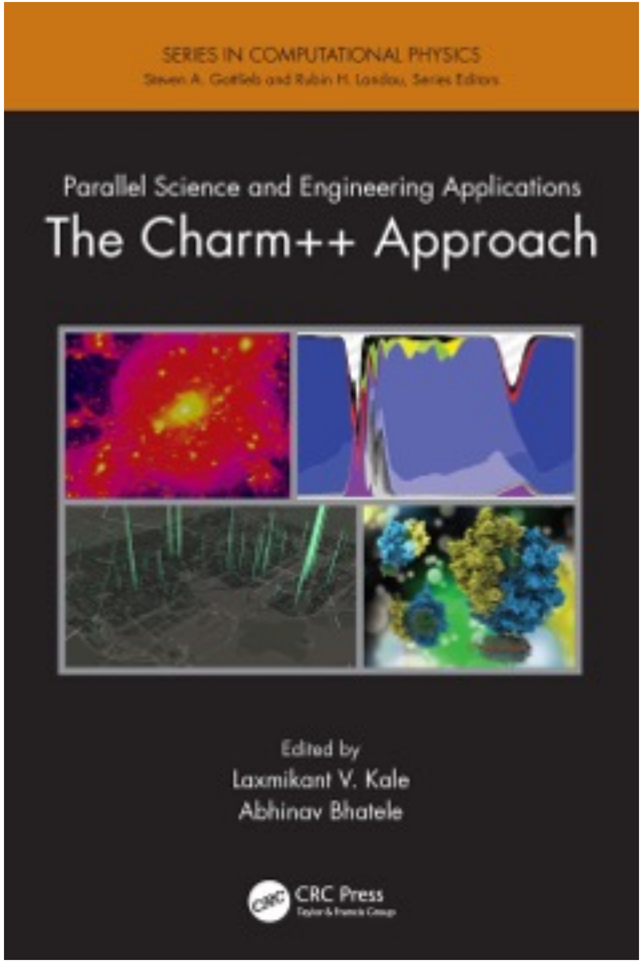
Anton
PITTSBURGH SUPERCOMPUTING CENTER

R01-GM086749 U54-GM087519
R01-GM101048 P41-GM104601

NAMD is based on Charm++

KLAUS SCHULTEN
Center for the Physics of Living Cells
University of Illinois
2012 IEEE COMPUTER SOCIETY
SIDNEY FERNBACH
MEMORIAL AWARD
LAXMIKANT V. KALE
National Center for Supercomputing
Applications · University of Illinois

Complete info at charmplusplus.org



SERIES IN COMPUTATIONAL PHYSICS
Steven A. Gottlieb and Rubin H. Landau, Series Editors

Parallel Science and Engineering Applications
The Charm++ Approach

Edited by
Laxmikant V. Kale
Abhinav Bhatele

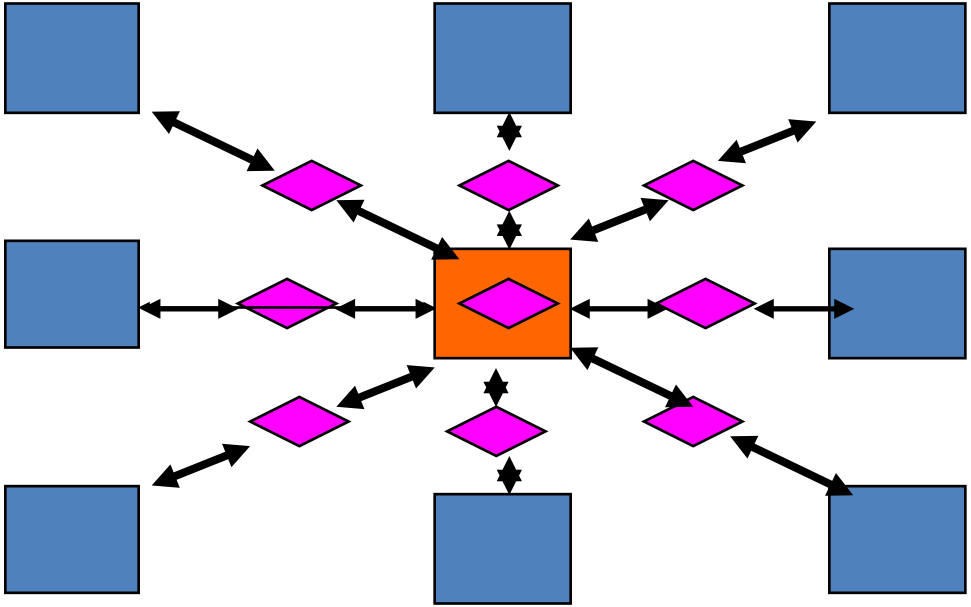
CRC Press
Taylor & Francis Group

Charm++ Used by NAMD

- Parallel C++ with *data driven* objects.
- Asynchronous method invocation.
- Prioritized scheduling of messages/execution.
- Measurement-based load balancing.
- Portable messaging layer.

NAMD Hybrid Decomposition

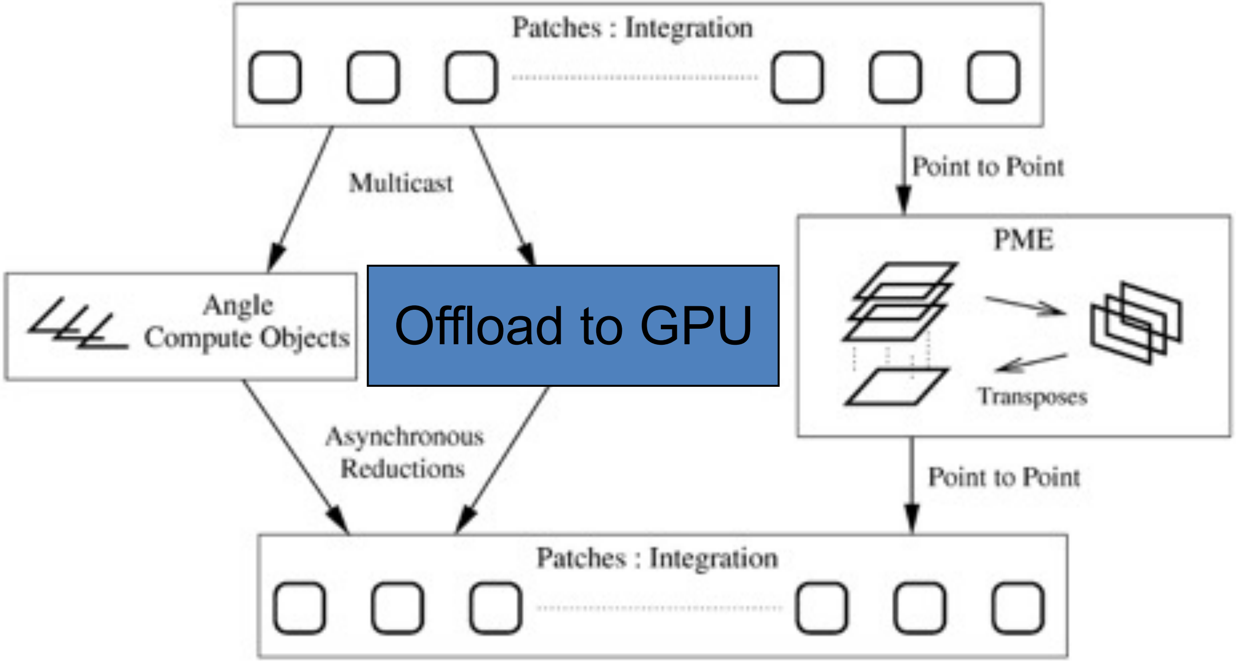
Kale *et al.*, *J. Comp. Phys.* 151:283-312, 1999.



- Spatially decompose data and communication.
- Separate but related work decomposition.
- “Compute objects” facilitate iterative, measurement-based load balancing system.

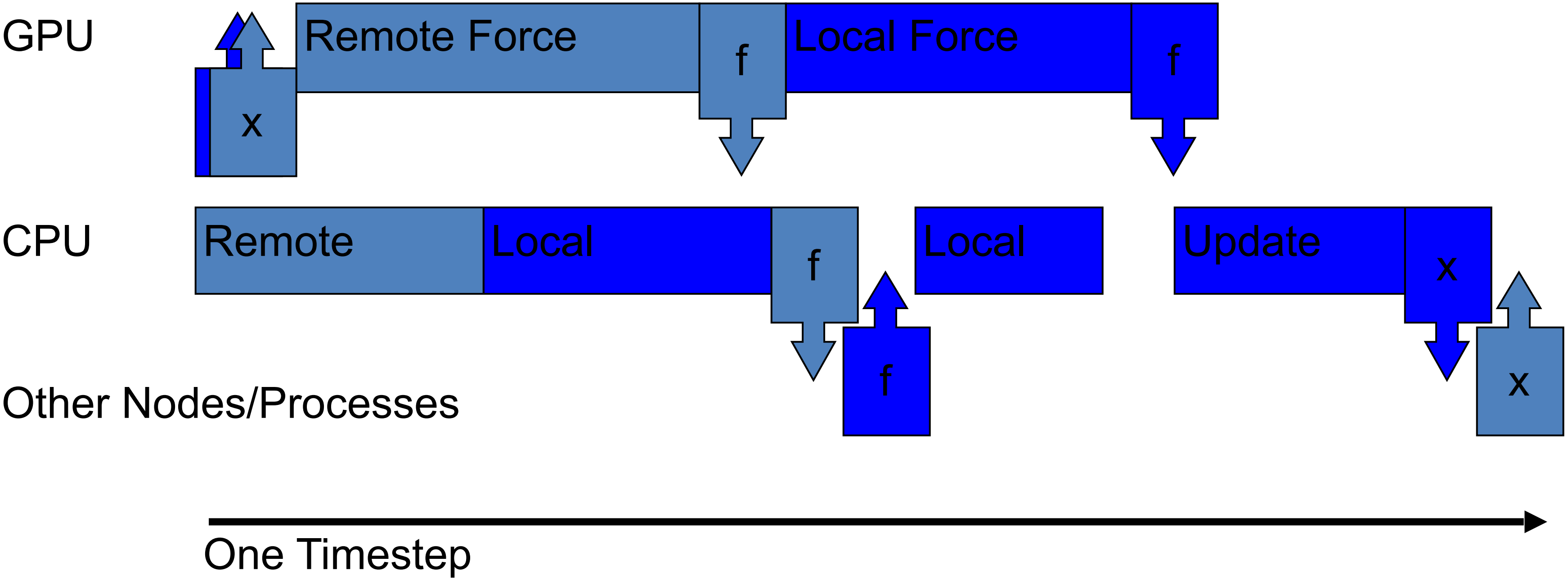
NAMD Overlapping Execution

Phillips *et al., SC2002*.



Objects are assigned to processors and queued as data arrives.

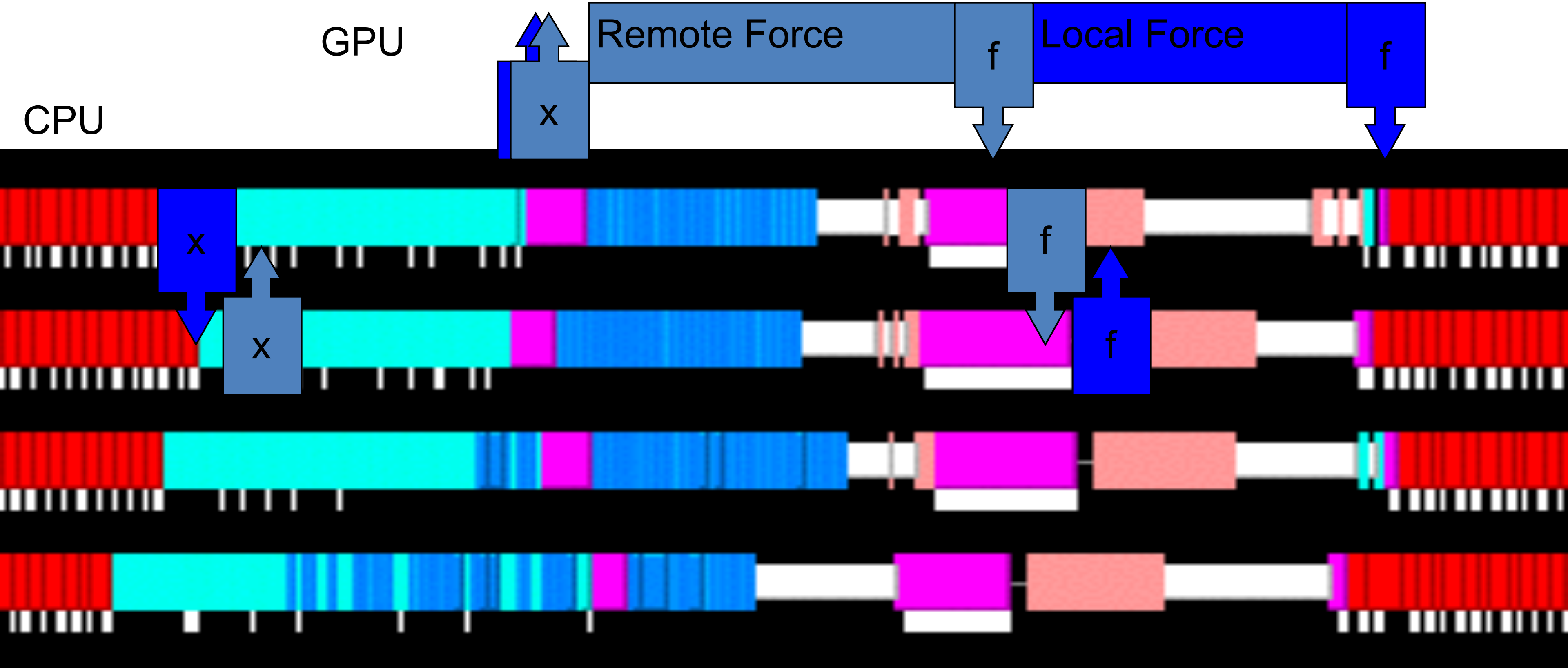
Overlapping GPU and CPU with Communication

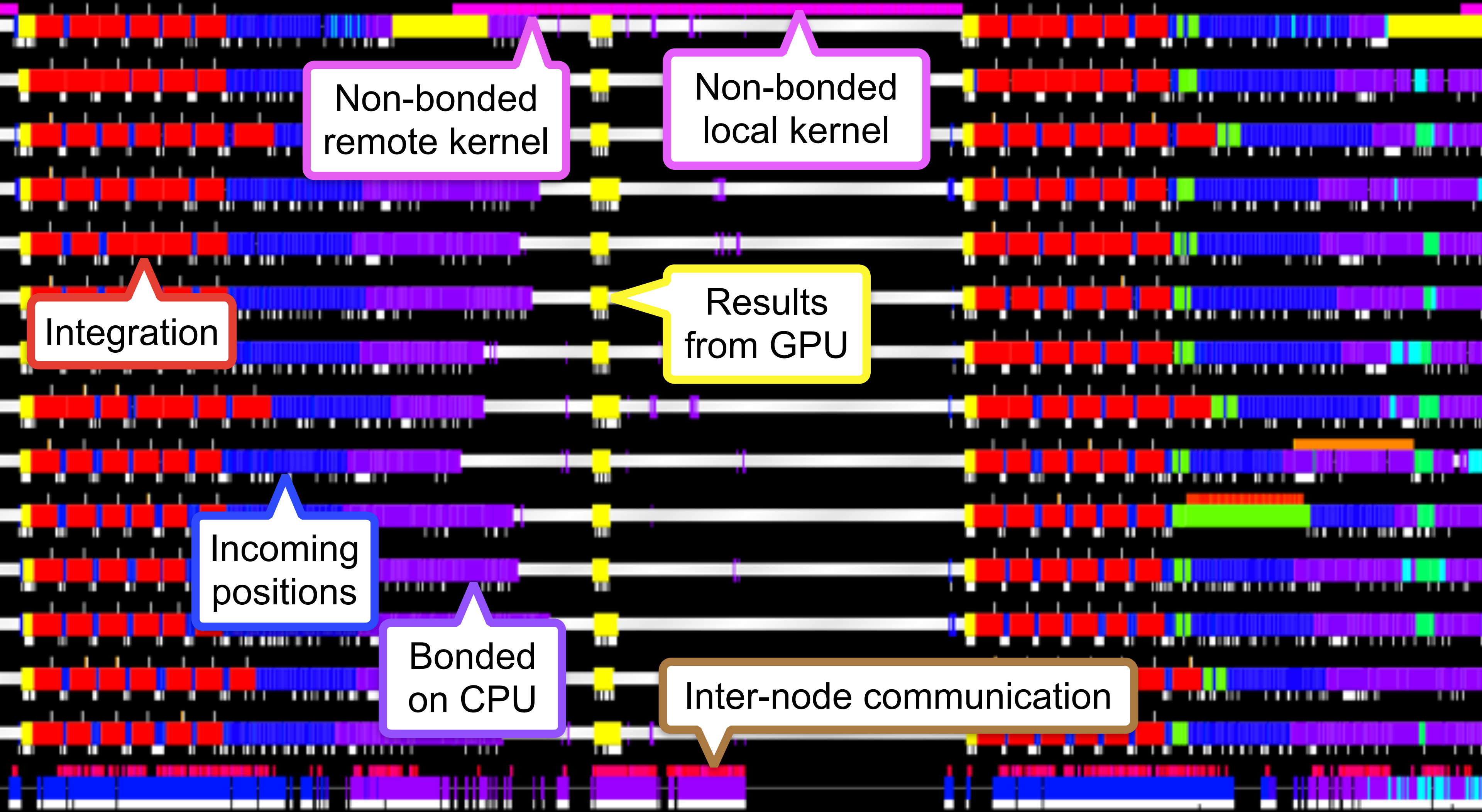


Phillips *et al.*, SC2008

Actual Timelines from NAMD

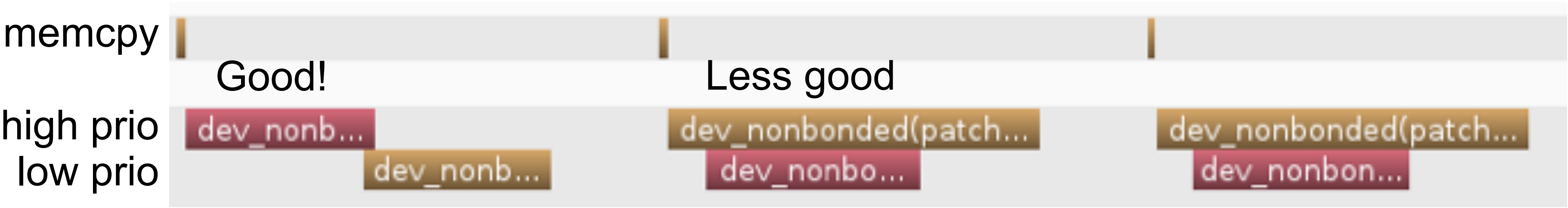
Generated using Charm++ tool “Projections” http://charm.cs.uiuc.edu/




Non-bonded remote kernel
Non-bonded local kernel
Results from GPU
Integration
Incoming positions
Bonded on CPU
Inter-node communication

Enabling Remote/Local Overlap

- Asking for priorities since 2008
  - Critical for Charm++ performance on CPU
  - With Kepler we get 1 bit on Tesla/Quadro
- Doesn’t order grid launches:

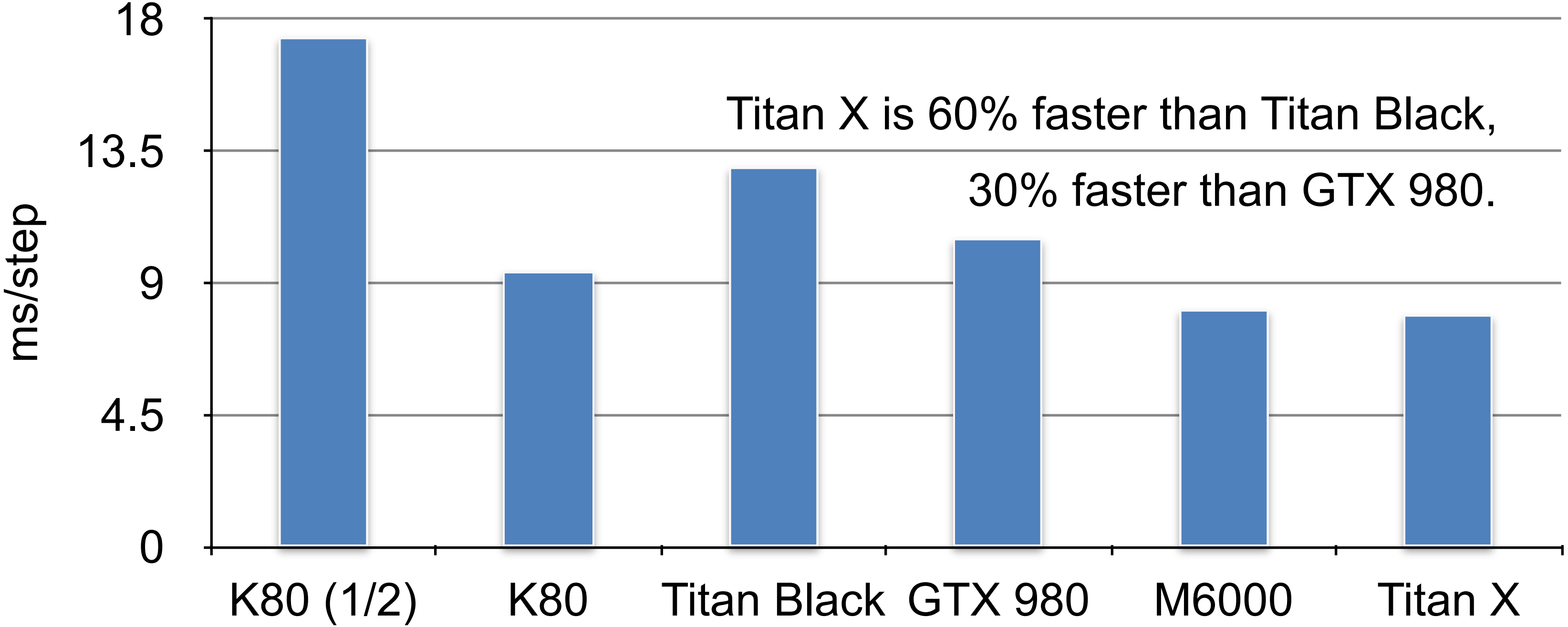


- Workaround is small memset in low-priority stream
  - Doesn’t need priorities, so works on GeForce cards too!

Kepler Shuffle Instructions

- Reductions for energy and pressure tensor
- Old implementation limits synchronization:
  - Reduce multiple fields at same time
  - Warp-synchronous for final stages
- Shuffle implementation is simpler and faster!
  - Except now preprocessor code for older devices
  - “diff –D KEPLER_SHUFFLE” is very helpful

Maxwell Performance



Titan X is 60% faster than Titan Black,
30% faster than GTX 980.

NAMD ApoA1 benchmark on 14 cores 2.6 GHz E5-2650 v2 or E5-2660 v3

CUDA 7

- We’ve heard of it.
- Looking forward to C++11.
- Runtime compilation might be awesome.
- We’ve also heard of CUDA 6.5.
- It will be available on Cray XK7 “soon”.
- Until then we’re stuck with CUDA 5.5.

Trends Affecting Performance

- GPU performance increasing
  - Performance limit will be code on CPU
  - Most highly tuned CPU code moved to GPU
  - Remaining CPU code is also less efficient
  - Therefore CPU must run serial code well
- CPU serial performance static
- CPU core counts increasing

Suggested Strategy

- Focus on CPU-side code
  - Port to GPU or optimize/paralellize on CPU
  - Stream results off GPU to increase overlap
  - Use CPUs with best single-thread performance
- Focus on communication
  - Reduce communication overhead on CPU
  - General parallel scalablity improvements
  - Map decomposition to machine torus topology
    - Also applies to replica exchange partitions

Phillips *et al.*, SC14

Torus Adaptation

- Job partitioning for multiple copy sampling algorithms
- Mapping NAMD spatial decomposition domains onto machine torus
- Mapping particle-mesh Ewald (PME) electrostatics onto spatial decomposition

Additional Techniques

- Coarsening of PME grid to reduce long-range communication
- Offloading of PME interpolation onto GPUs
- Removal of implicit synchronization in pressure control algorithm

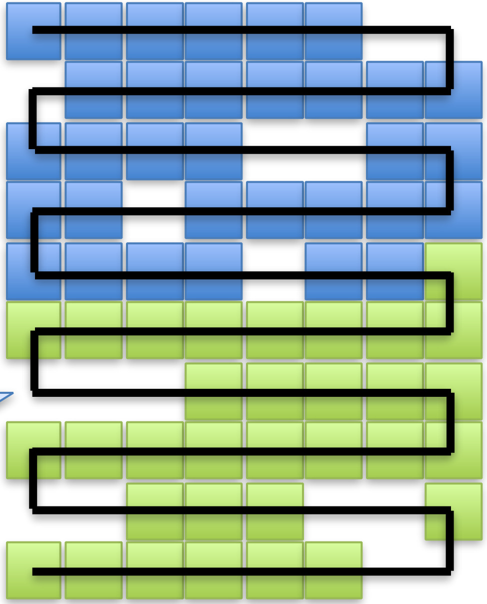
Irregular Torus Topologies

- IBM Blue Gene L/P/Q provide jobs with complete, regular, power-of-two torus.
- Cray XE/XK job topology is unpredictable.
  - – Scheduler works around already running jobs.
  - – May not be compact or contiguous.
    - New Blue Waters scheduler addresses this.
  - – Even full-machine jobs skip over I/O nodes.

Convert Torus to Optimized Mesh

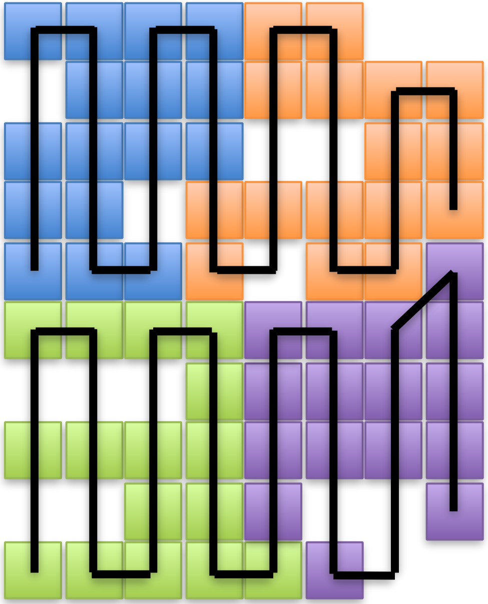
- Start with Charm++ TopoManager API
  - Provides node coordinates and torus dimensions.
- Extend with TopoManagerWrapper class
  - Ensure same torus coordinates for entire physical node.
  - Shift torus coordinates to eliminate largest gap in node list.
  - Re-order dimensions from longest to shortest occupied span.
  - Provide functions for sorting list of ranks along ordered list of dimensions by “snake scanning” curve (seen on next slide).
- Recursive bisection on these “snake scanning” curves is the basis of all torus-mapping algorithms to follow.

Mapping Charm++ Partitions



Gaps in torus

Bisection 1

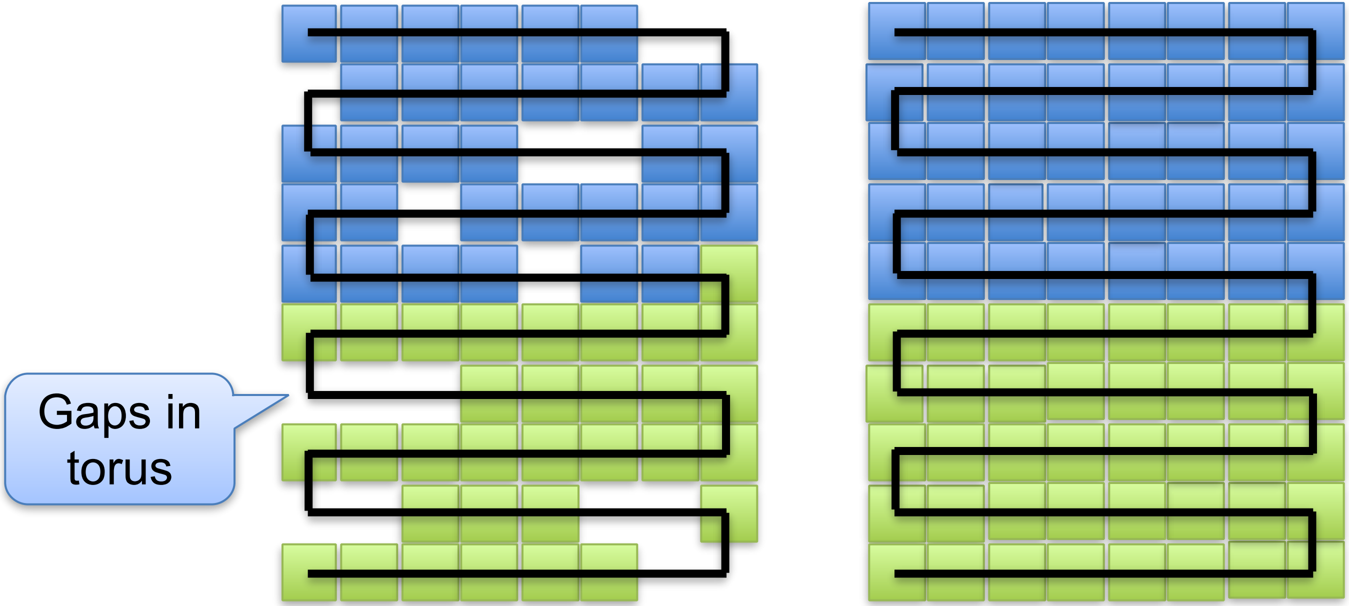


Bisection 2

Mapping NAMD Spatial Domains

- Priorities are:
  1. Evenly distributed patch load across available PEs
  2. Compact patch set within physical node to minimize communication
  3. Torus topology adaptation – only impacts largest runs
- Simultaneous recursive bisection of patch mesh and PE mesh:
  - Re-order patch and PE mesh dimensions longest to shortest.
  - When dividing PEs, divide patches along corresponding dimension, if possible, before falling back to next-longest dimension.
  - Divide PEs on physical node boundaries.
  - Divide patches to balance load with at least one patch per PE.
- Within physical node, sort patches along PME slabs/pencils.

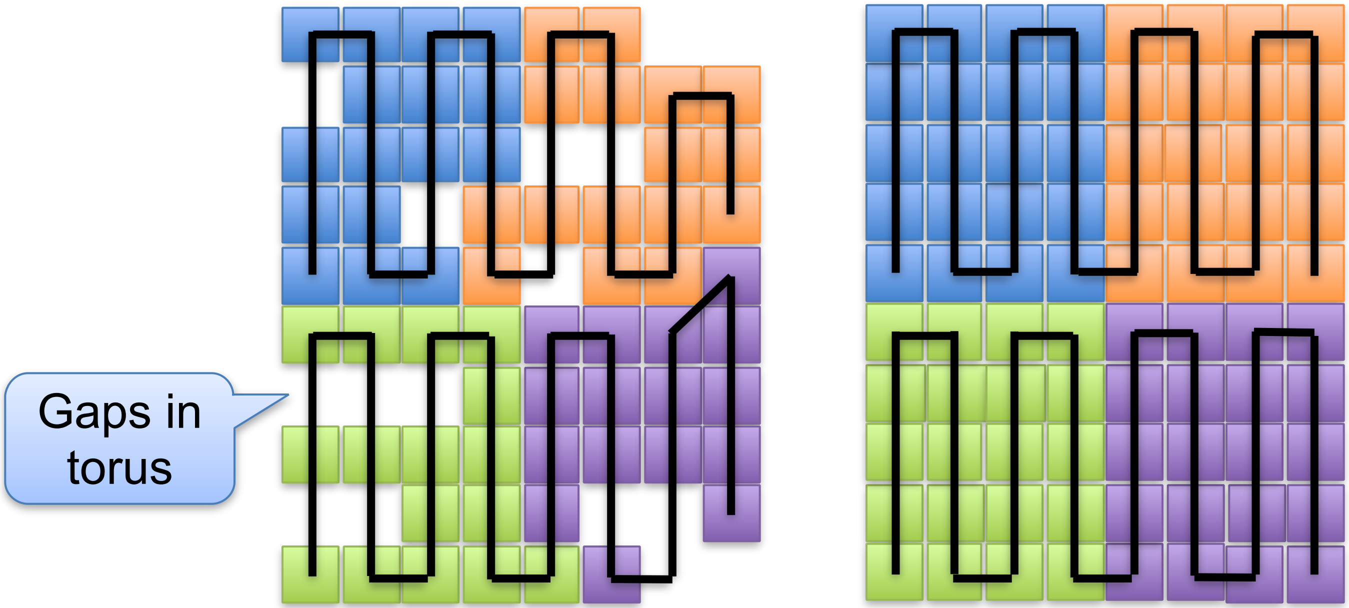
Mapping NAMD Spatial Domains



Processors

Patches

Mapping NAMD Spatial Domains



Gaps in torus

Processors

Patches

Mapping PME Electrostatics

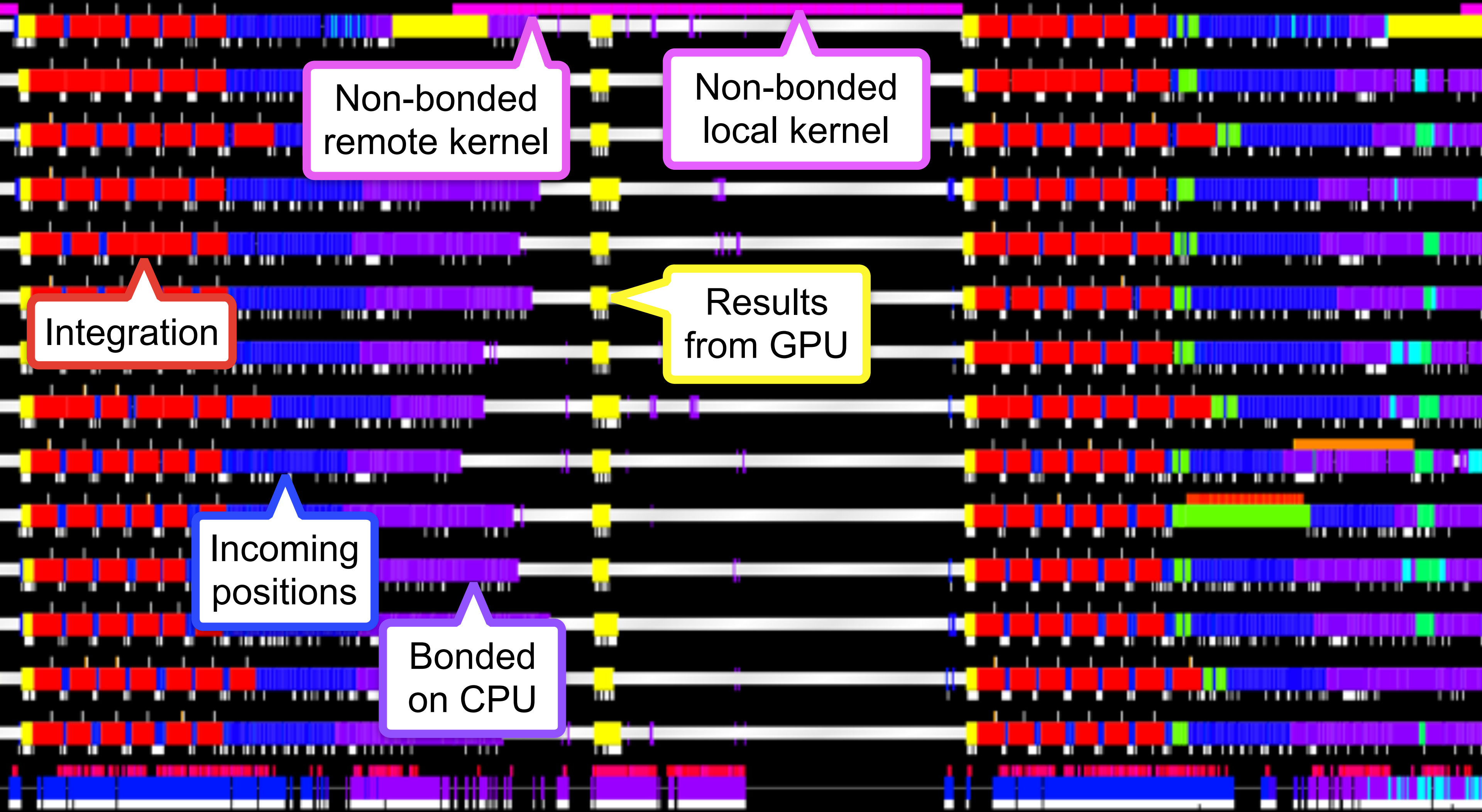
- Want to align X-Y grid of Z pencils to patches.
  - Needs to work even on non-torus machines.
- Assign X-Y coordinates to PEs.
  - Average coordinate of patches on PE (or node, etc.)
- Recursively bisect…
  - Z pencils on longer dimension boundary (5x5=2x5+3x5).
  - PEs proportionately (25=10+15) on same coordinate.
- Optimize Y-X-Y FFT transposes by placing X and Y pencils with same Z coordinate on contiguous ranks.

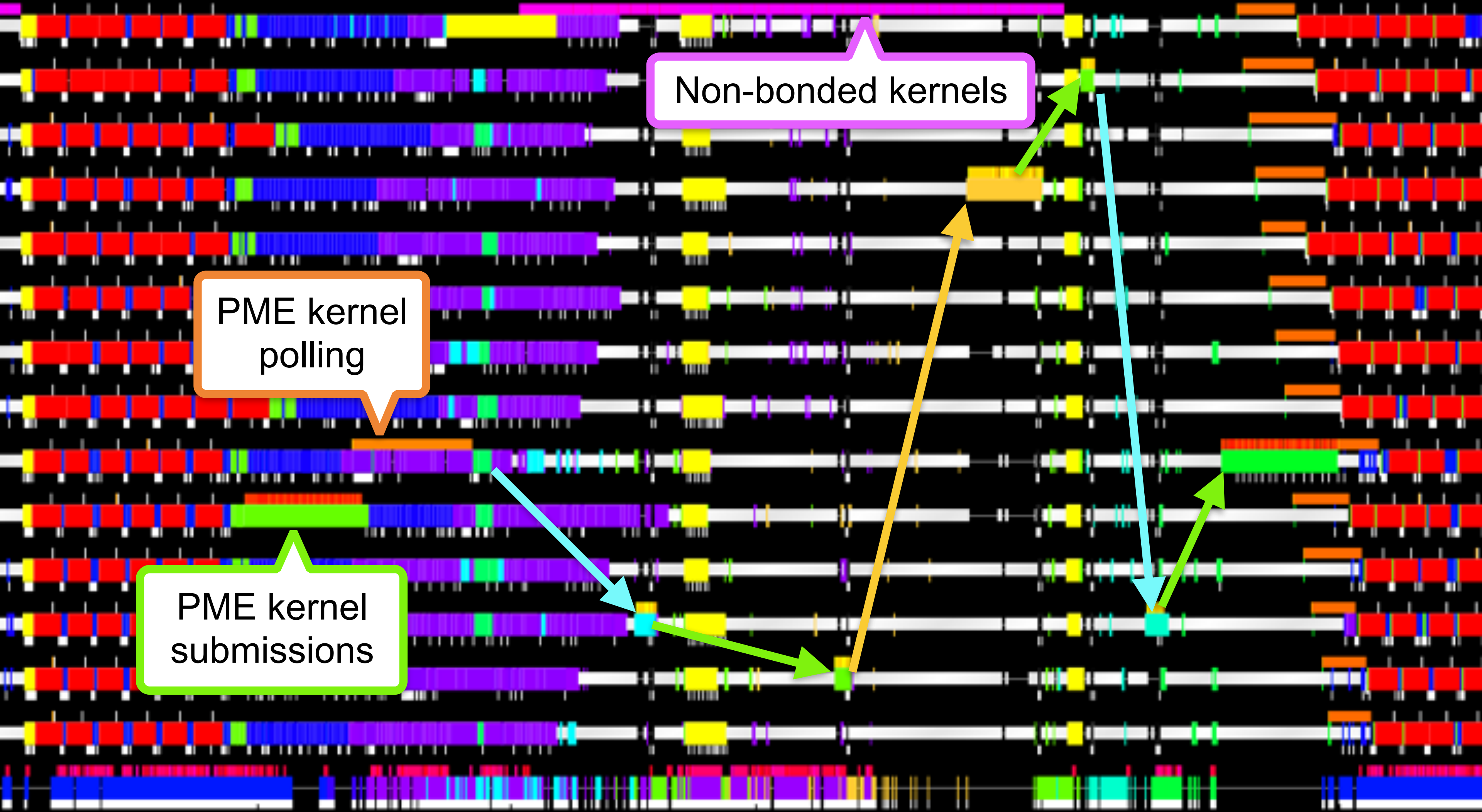
NAMD PME CUDA Kernel

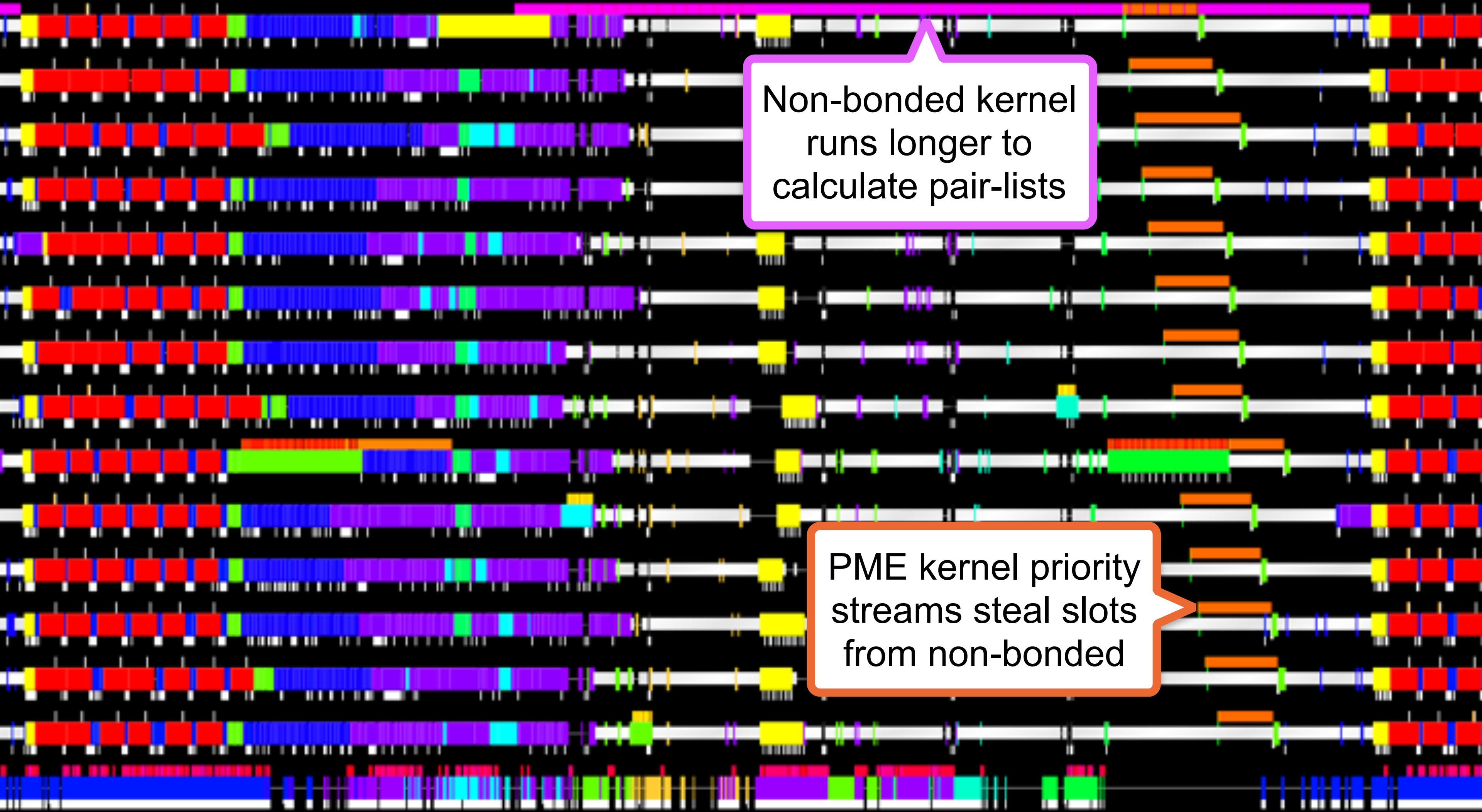
- Bottleneck for 100M atoms is PME FFT communication
  - Switch from 4th-order to 8th-order interpolation on coarser grid
- Doing 8th-order PME on GPU improves critical path
- CPU may be bottleneck for 8th-order PME
  - Especially as GPU non-bonded gets faster…
- Simplest design that might possibly work:
  - One stream per host PE (preserve control flow)
  - One atom per warp with warp-synchronous programming
  - Atomics to accumulate charge grid in global memory
    - One per thread so accesses coalesce
    - Also build “used” flags arrays for x-y pencils and z plane

PME Kernel Aggregation

- Initial version slower than PME on CPU
- First, one launch per PE, not per patch
- Second, one charge array per node
  - First version to beat PME on CPU
  - Node-level coordination a challenge in Charm++
  - Reduces number of messages sent per node!
    - Need to backport to PME on CPU version
    - May help CPU-only version, but not as much


Non-bonded remote kernel
Non-bonded local kernel
Results from GPU
Integration
Incoming positions
Bonded on CPU


Non-bonded kernels
PME kernel polling
PME kernel submissions



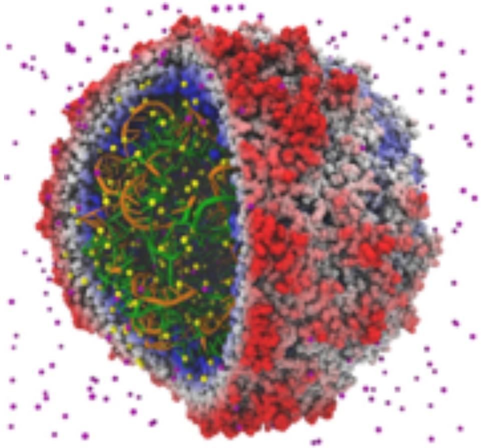
Non-bonded kernel runs longer to calculate pair-lists

PME kernel priority streams steal slots from non-bonded

Performance Results

- Petascale simulation preparation is not easy.
  - Benchmarks based on 1.06M-atom STMV
  - 5x2x2 grid = 21M atoms ~ “small petascale”
  - 7x6x5 grid = 224M atoms ~ “Influenza virus”

- Experiment by disabling optimizations
  - Only disable one at a time, **not cumulatively**.

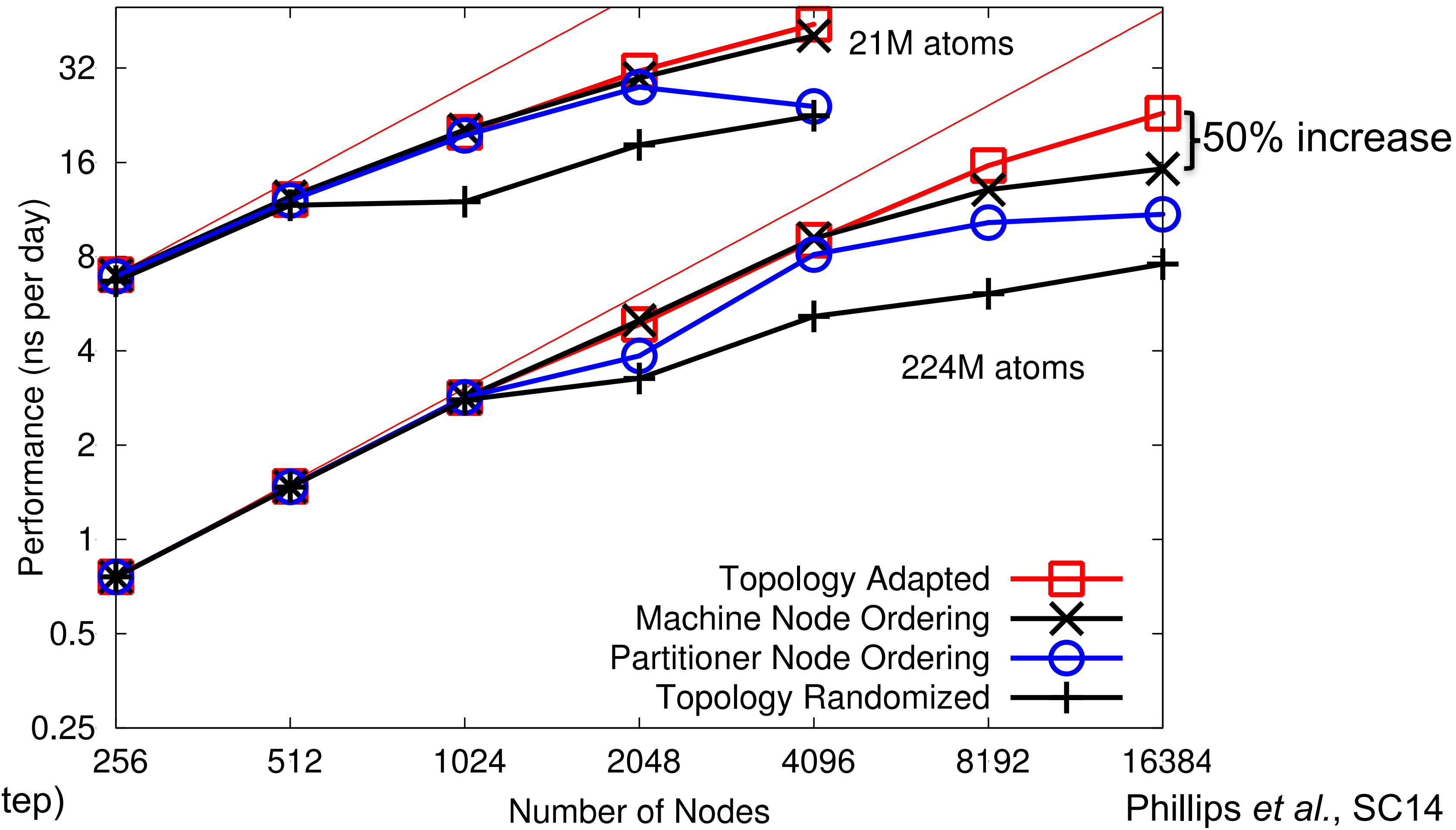


Huge system in 2006

Benchmarking Caution

- Cray XE/XK performance varies due to:
  - Compactness of nodes assigned to job
  - Other jobs running on machine (cross-traffic)
  - I/O activity (more Blue Waters than Titan)
- To test performance impact of changes, run old and new back-to-back in *same job*.

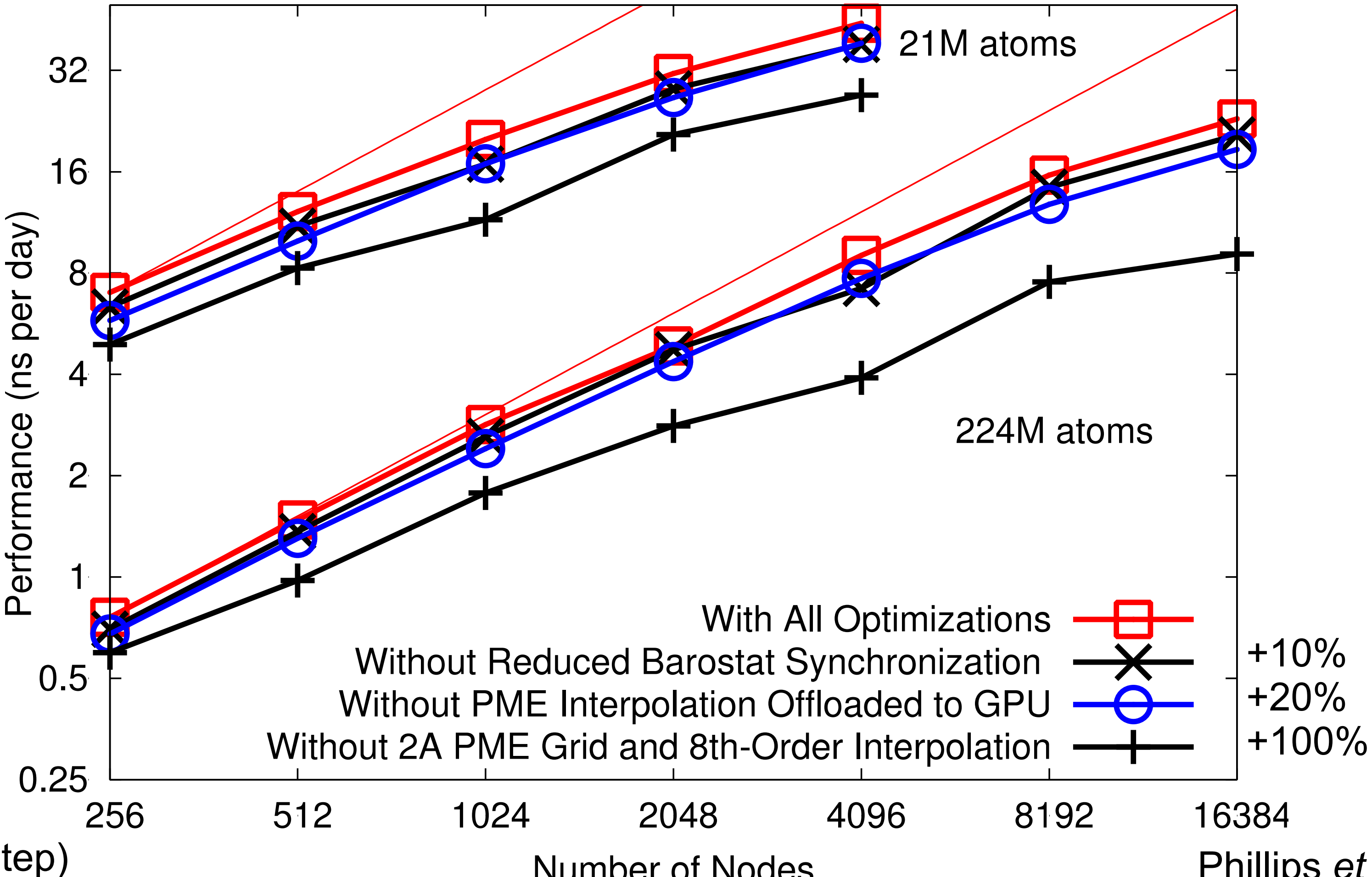
NAMD Topology Mapping on Titan Cray XK7



(2fs timestep)

Phillips *et al.*, SC14

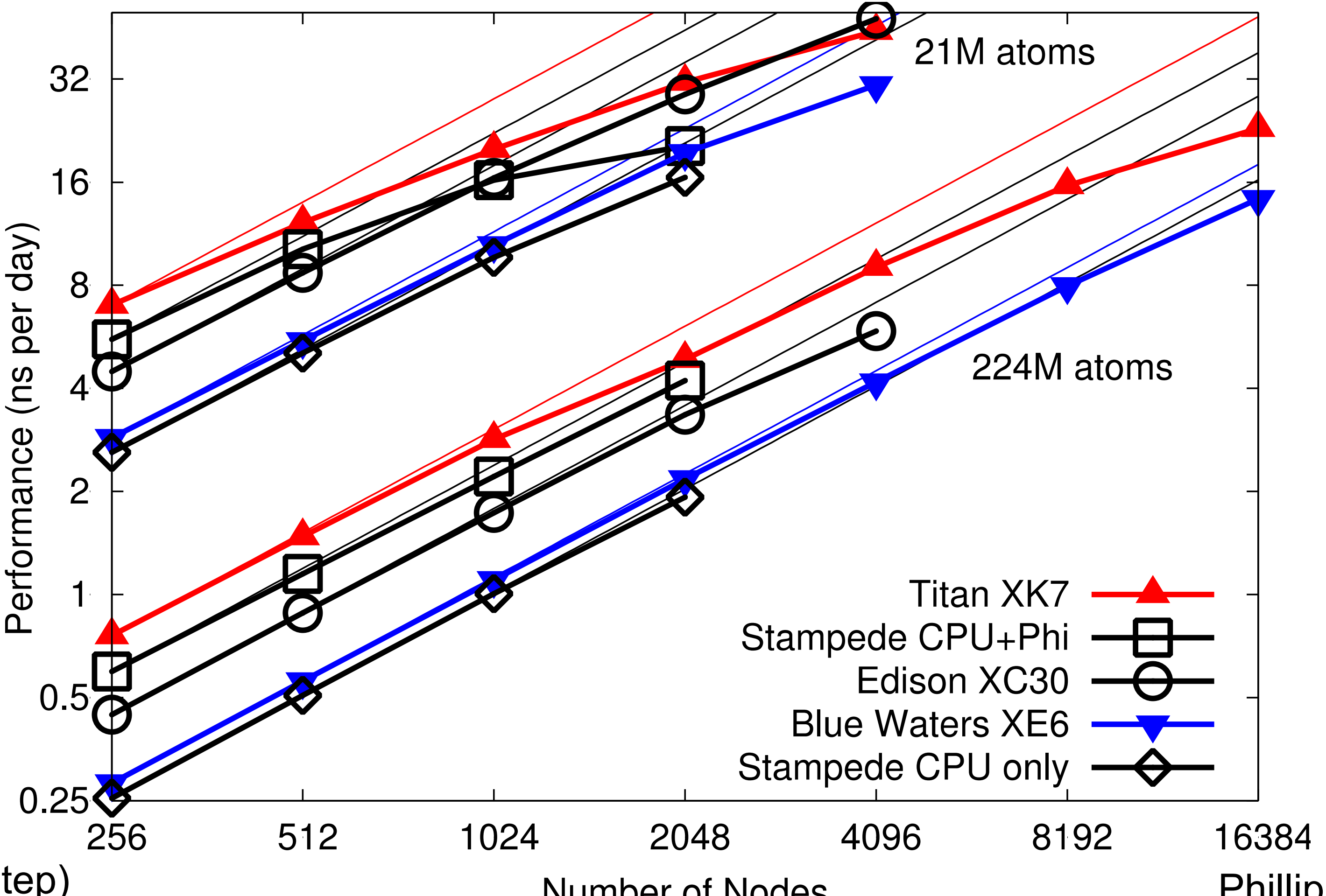
Other NAMD Optimizations on Titan Cray XK7



(2fs timestep)

Phillips *et al.*, SC14

NAMD on Torus and Non-torus Networks



(2fs timestep)

Phillips *et al.*, SC14

Streaming CPU Results to CPU

- Allows incremental results from a single grid to be processed on CPU before grid finishes on GPU
- Allows merging and prioritizing of remote and local work
- GPU side:
  - Write results to host-mapped memory (also without streaming)
  - __threadfence_system() and __syncthreads()
  - Atomic increment for next output queue location
  - Write result index to output queue
- CPU side:
  - Poll end of output queue (int array) in host memory

Streaming on GPU:

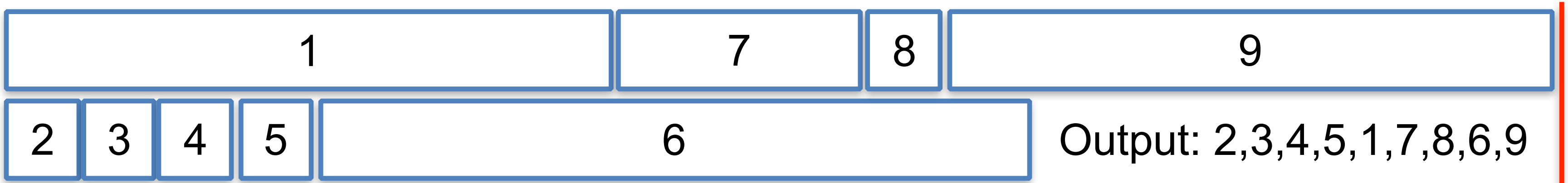
```
if ( force_ready_queue ) {
        __threadfence_system();
        __syncthreads();
        if (threadIdx.x == 0) {
          int old = atomicInc(force_list_counters,force_lists_size-1);
          force_ready_queue[old] = myPatchPair.patch1_force_list_index;
          __threadfence_system();
        }
}
```

Polling on host:

```
while ( -1 != (flindex = force_ready_queue[force_ready_queue_next]) ) {
        force_ready_queue[force_ready_queue_next] = -1;
        ++force_ready_queue_next;
        …process output flindex…
 }
```

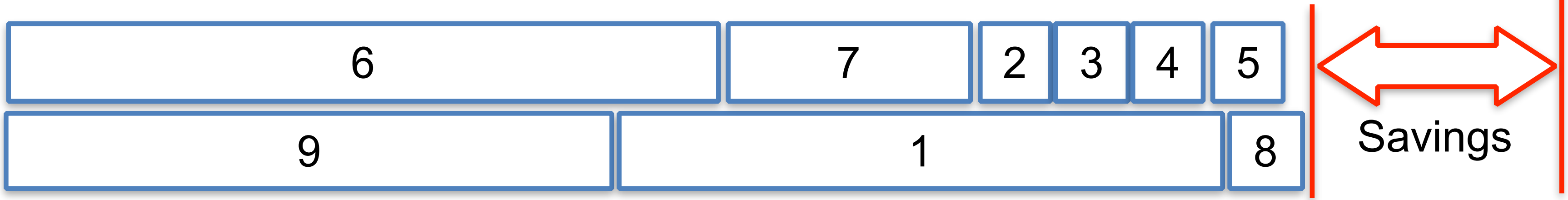
Controlling Output Order

- Blocks have widely varying runtimes
  - Input order is not output order



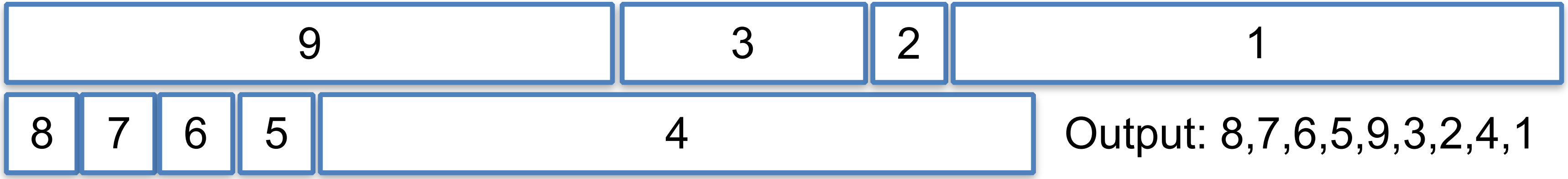
Output: 2,3,4,5,1,7,8,6,9

- Non-streaming was simple, just sort large to small



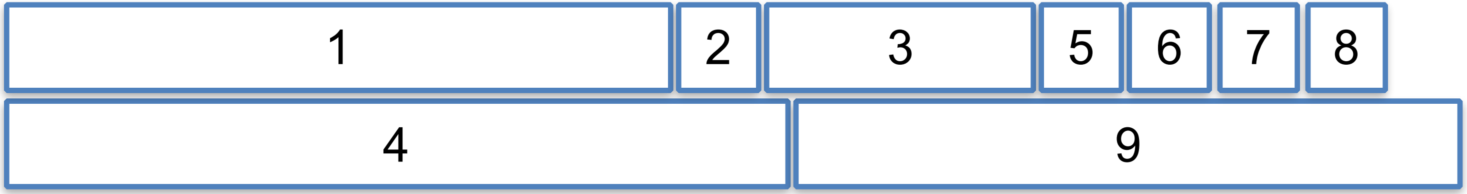
Controlling Output Order

- First use reversed priorities as input order



Output: 8,7,6,5,9,3,2,4,1

- Then reverse output order to use as input



- Provides good ordering and near-ideal compactness

Controlling Output Order

- Requires very little code to save order

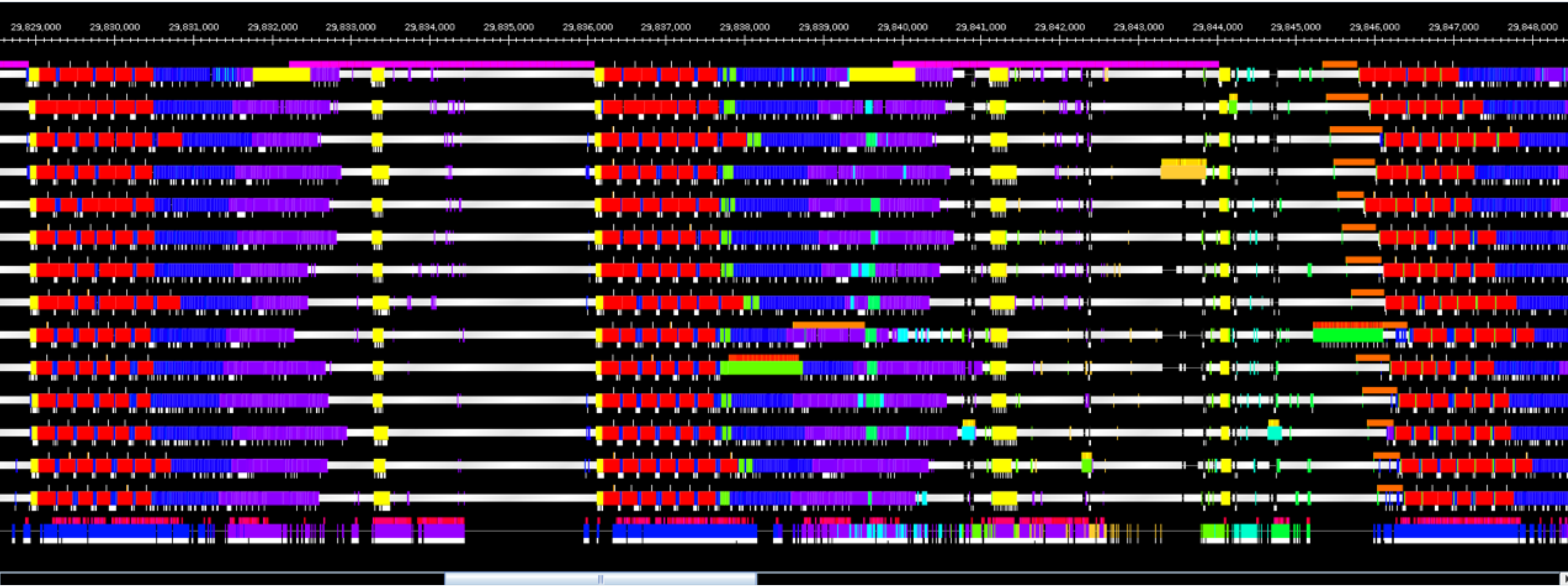
```
if (threadIdx.x == 0 && block_order) {
        int old = atomicInc(force_list_counters+1,total_block_count-1);
        block_order[old] = block_begin + blockIdx.x;
}
```

- Does not require measuring block runtimes
- Better than old heuristic ordering
- Streaming wins even on single node!

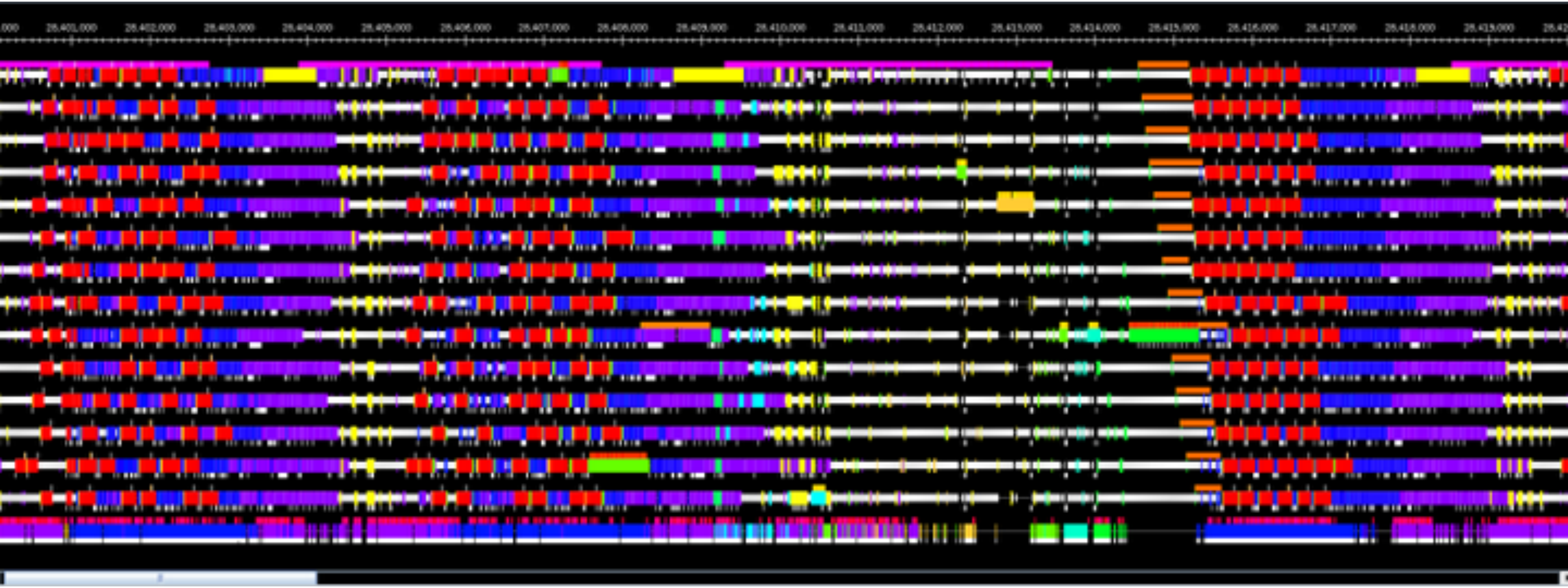
Controlling Output Order

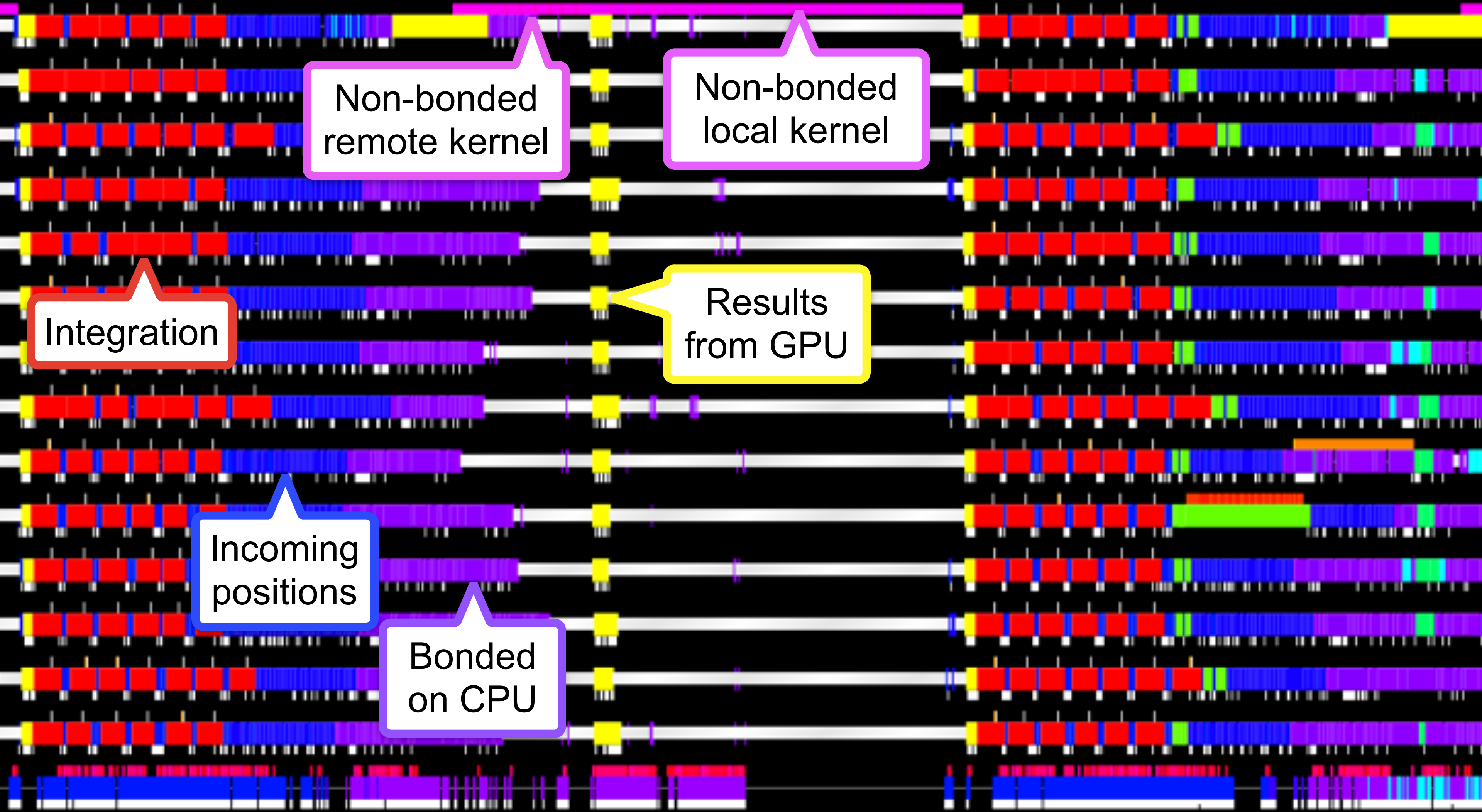
- But what is optimal output order?
  - Remote before local (same as before)
  - Distribute local across threads
    - Slight preference for GPU host thread
  - Local without remote proxies last
    - Not yet implemented

Non-Streaming Kernel

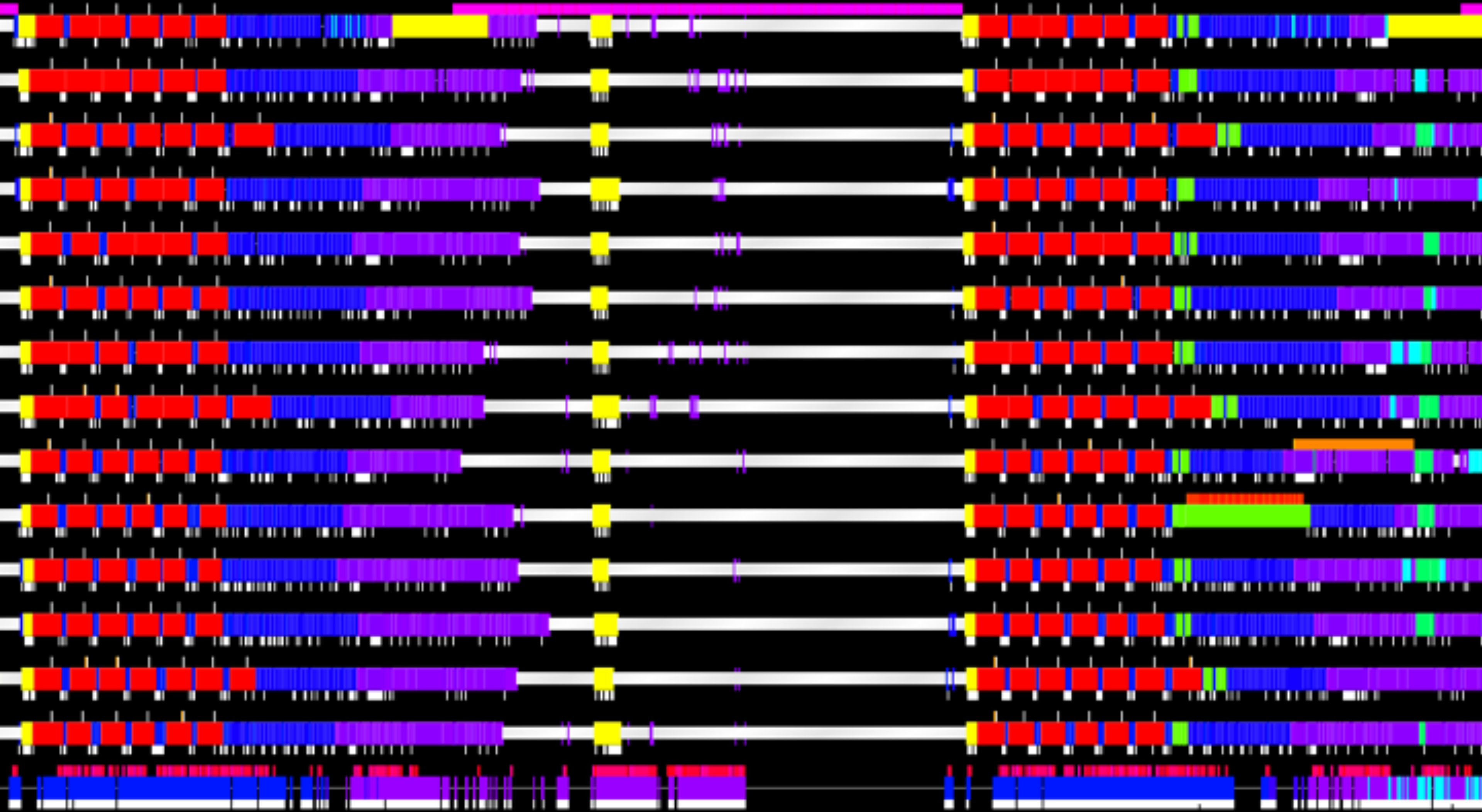


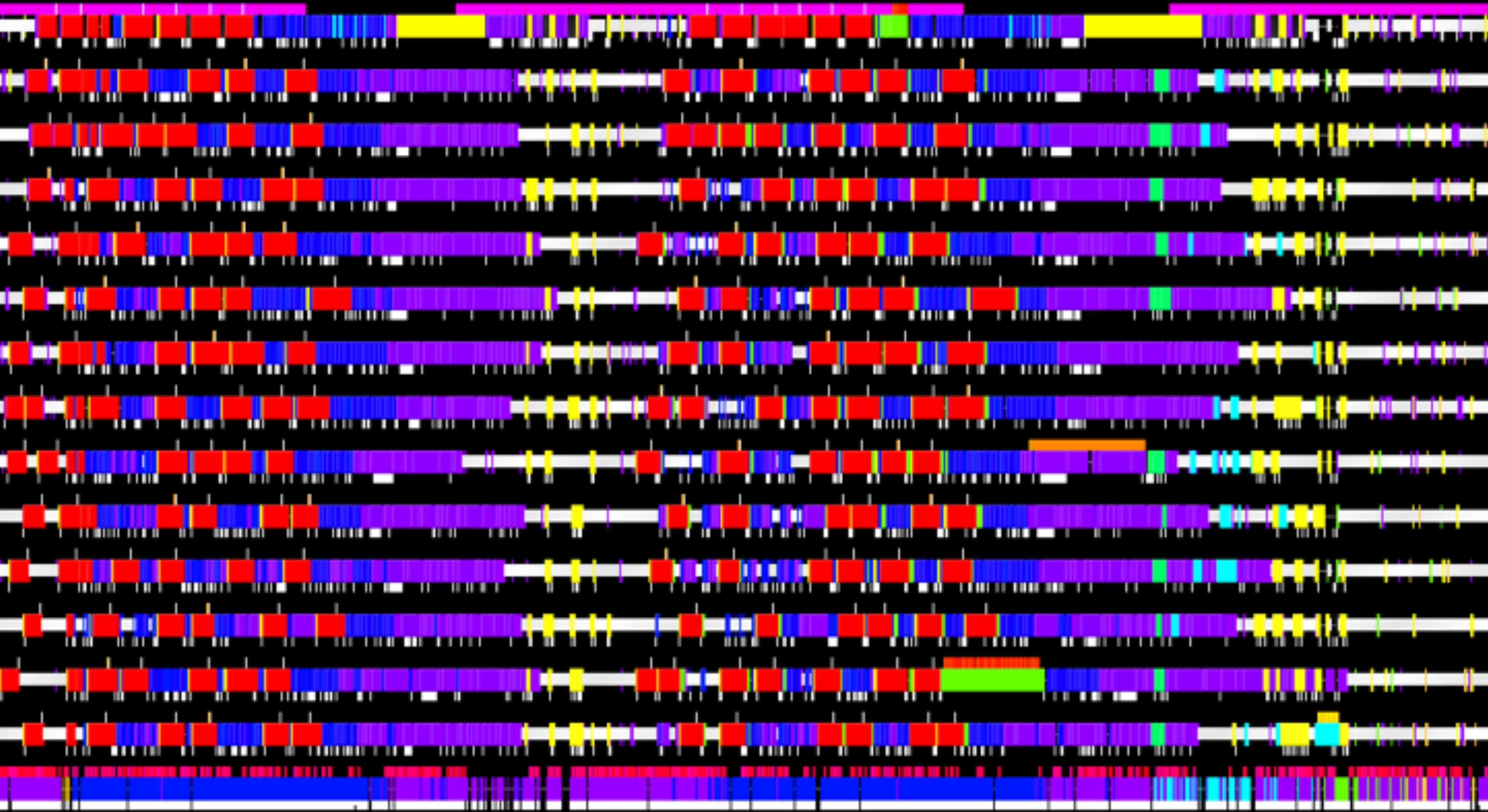
Streaming Kernel

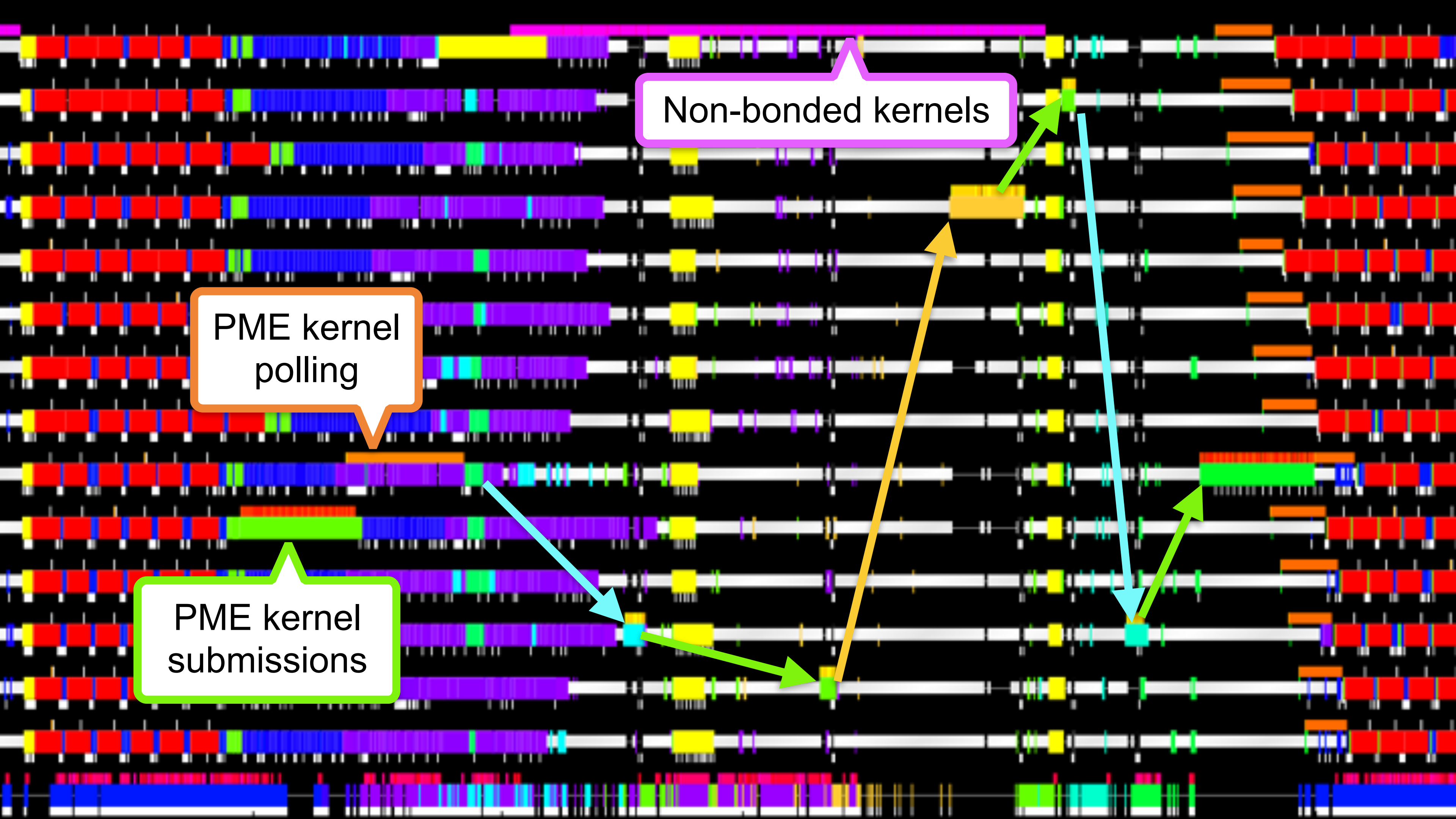


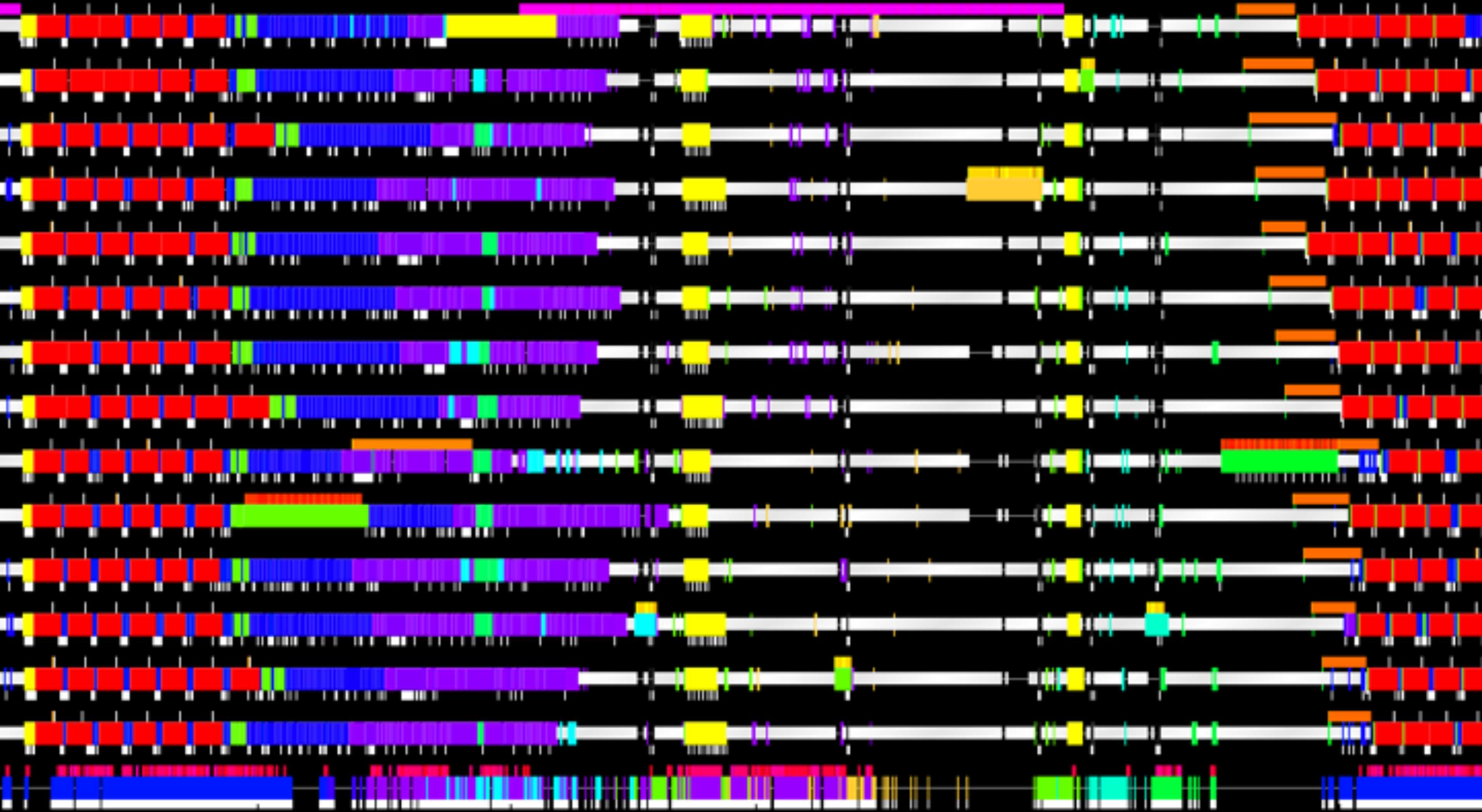


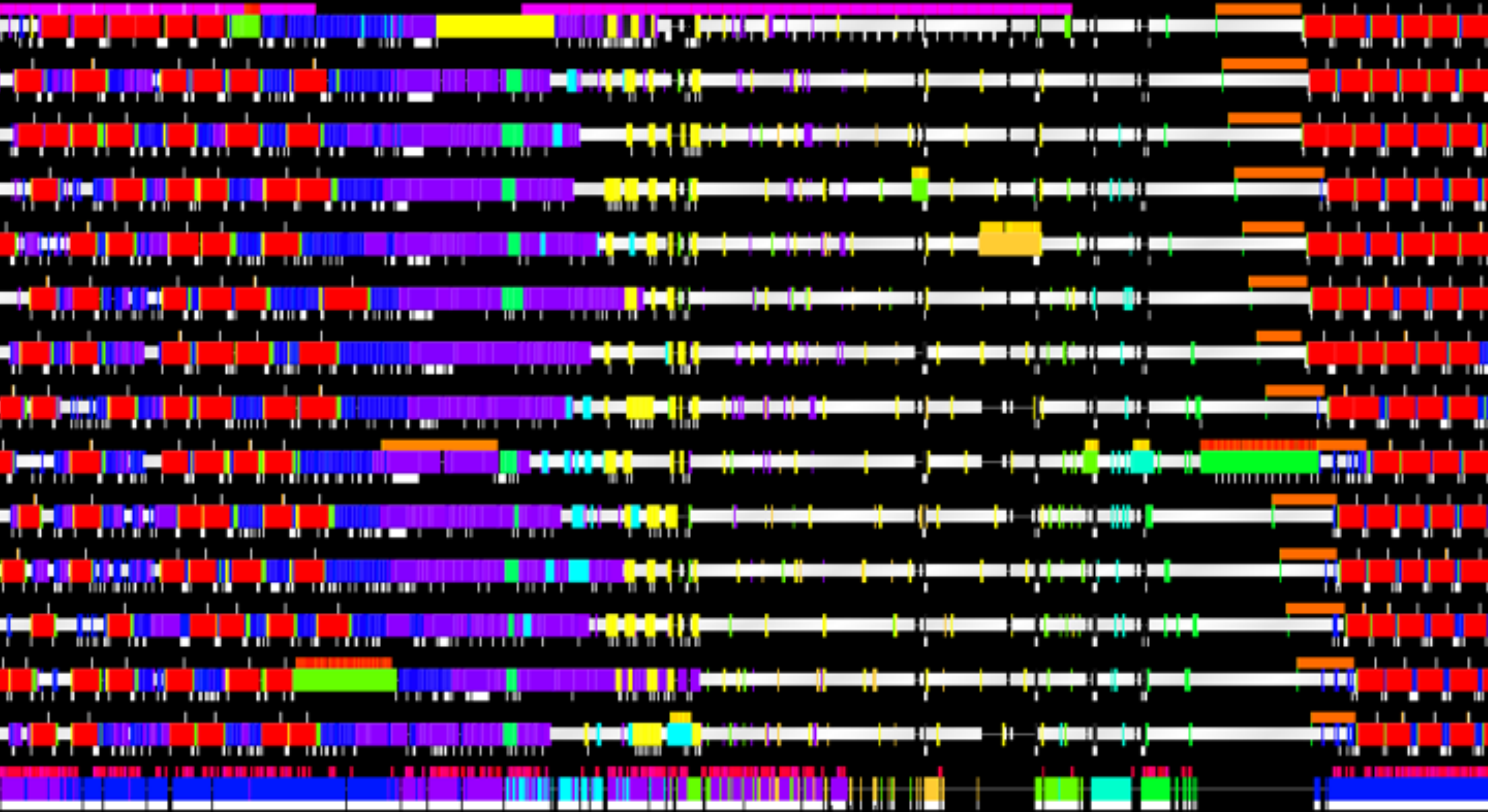
Non-bonded remote kernel
Non-bonded local kernel
Integration
Results from GPU
Incoming positions
Bonded on CPU



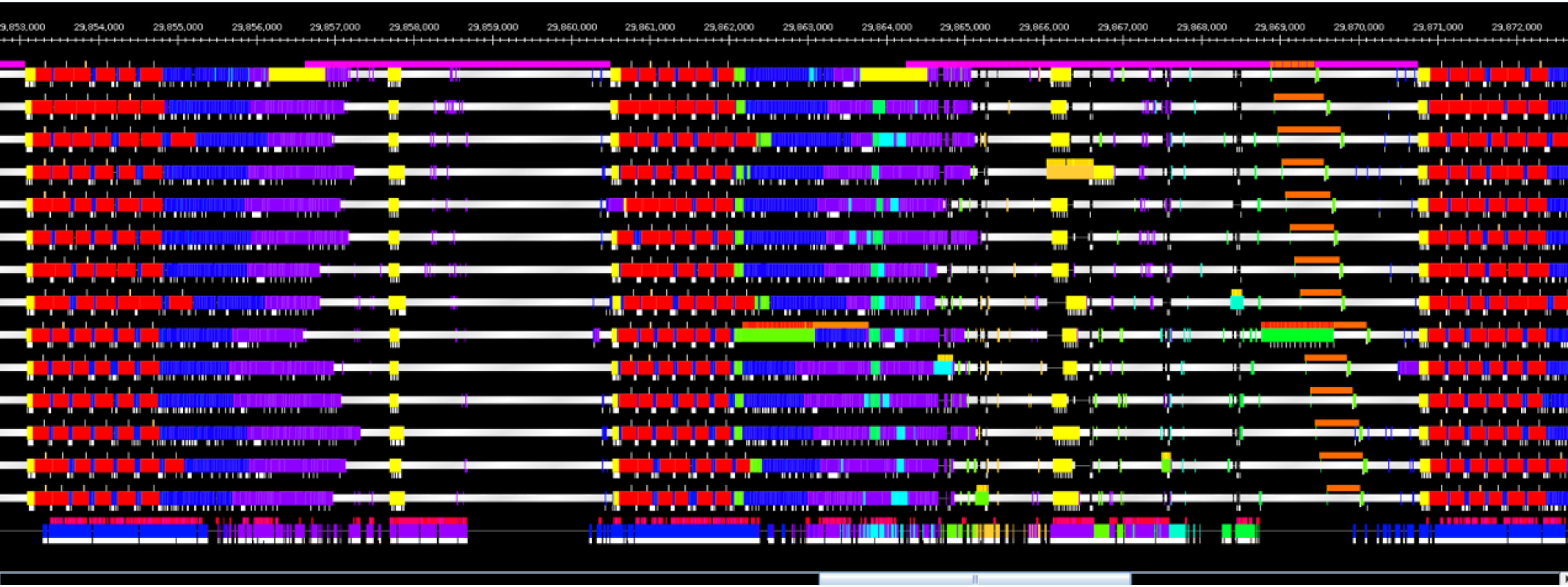



Non-bonded kernels
PME kernel polling
PME kernel submissions

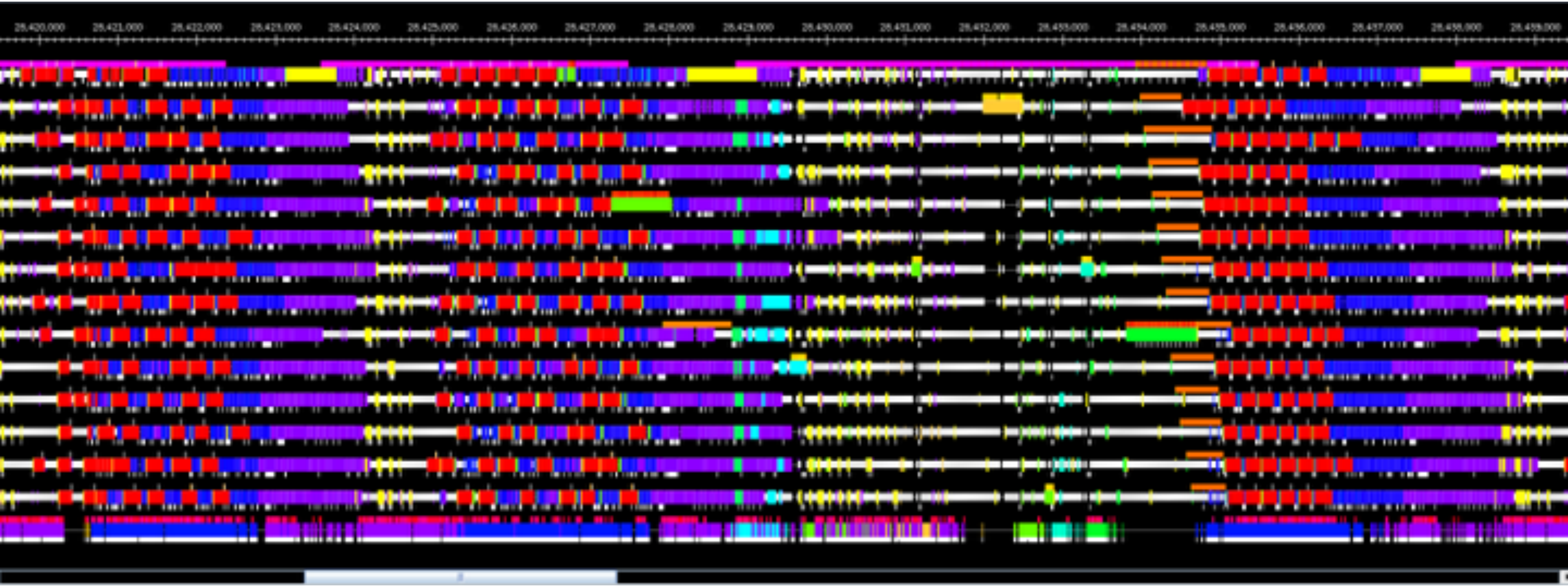




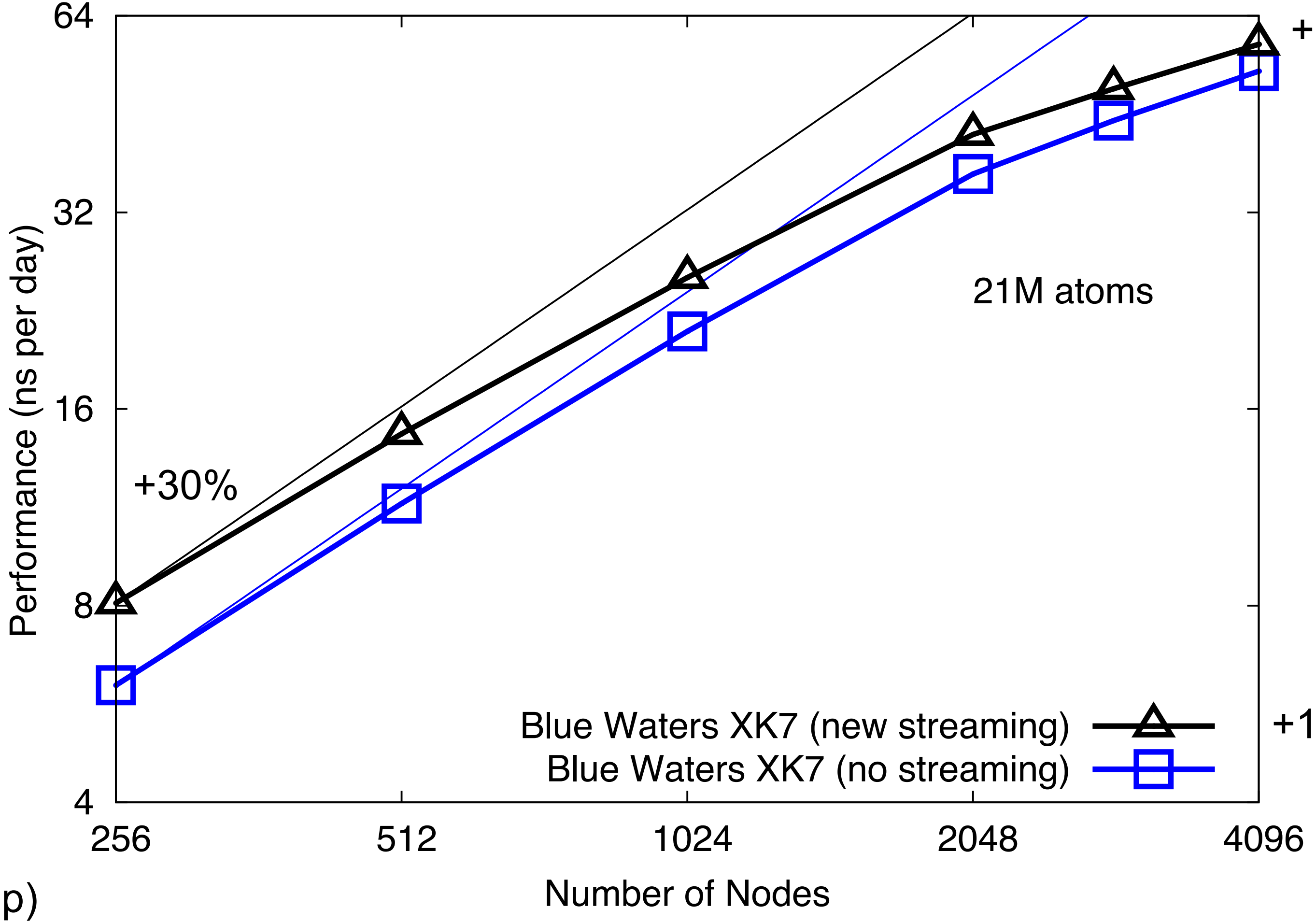
Non-Streaming Kernel Pairlist Step



Streaming Kernel Pairlist Step



New Streaming Kernel Performance



+10%

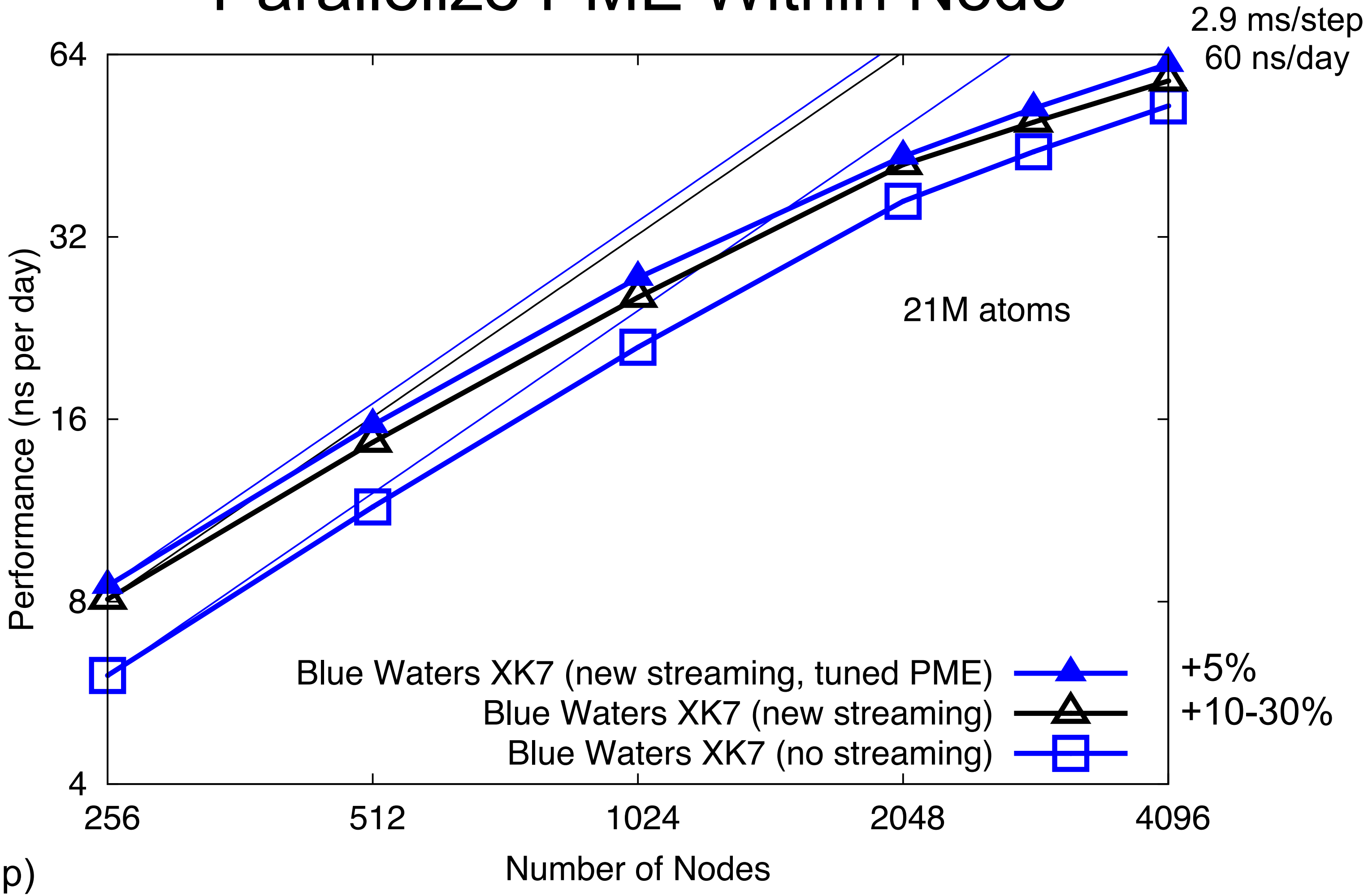
+30%

21M atoms

+10-30%

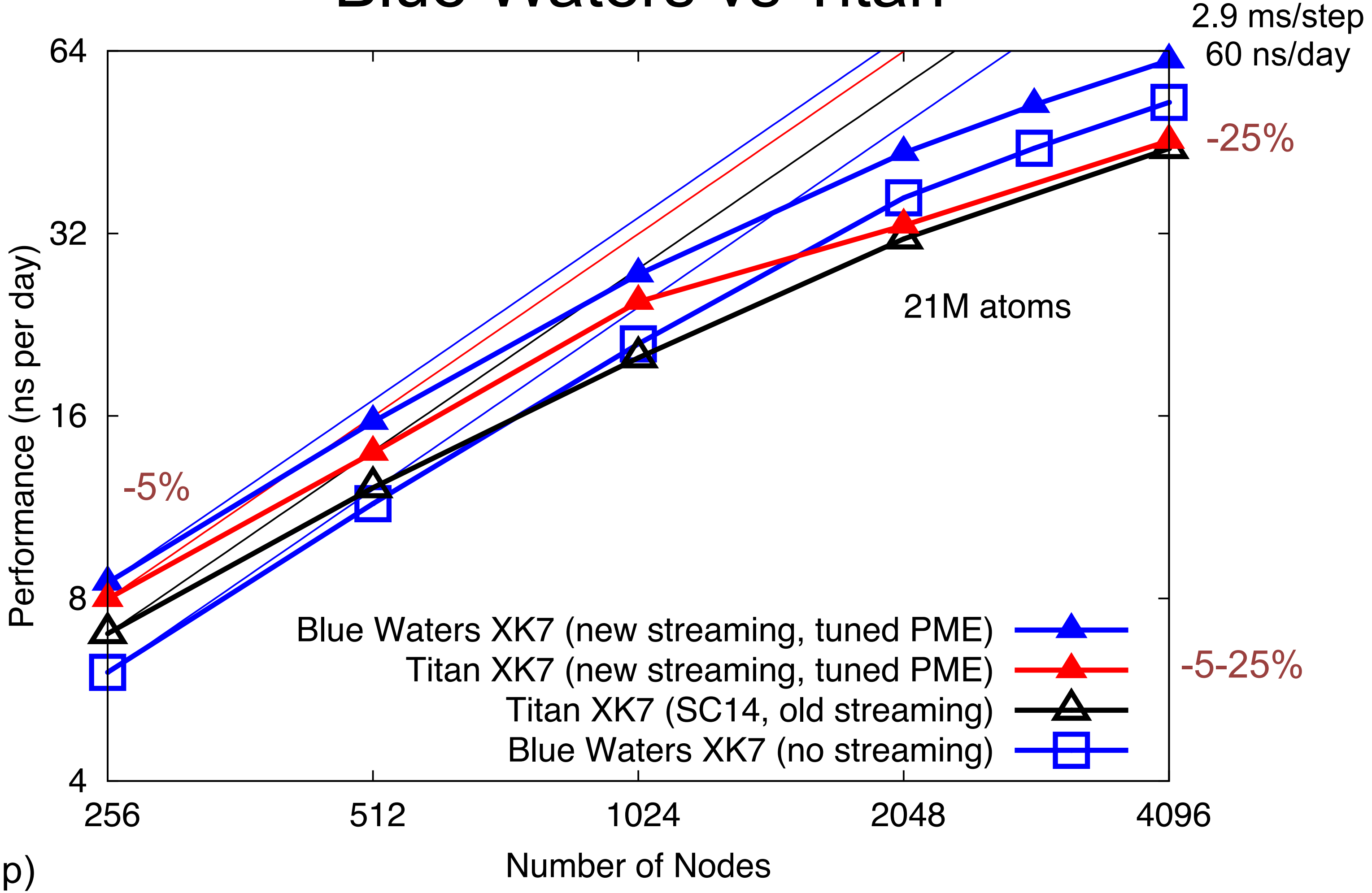
(2fs timestep)

Parallelize PME Within Node



(2fs timestep)

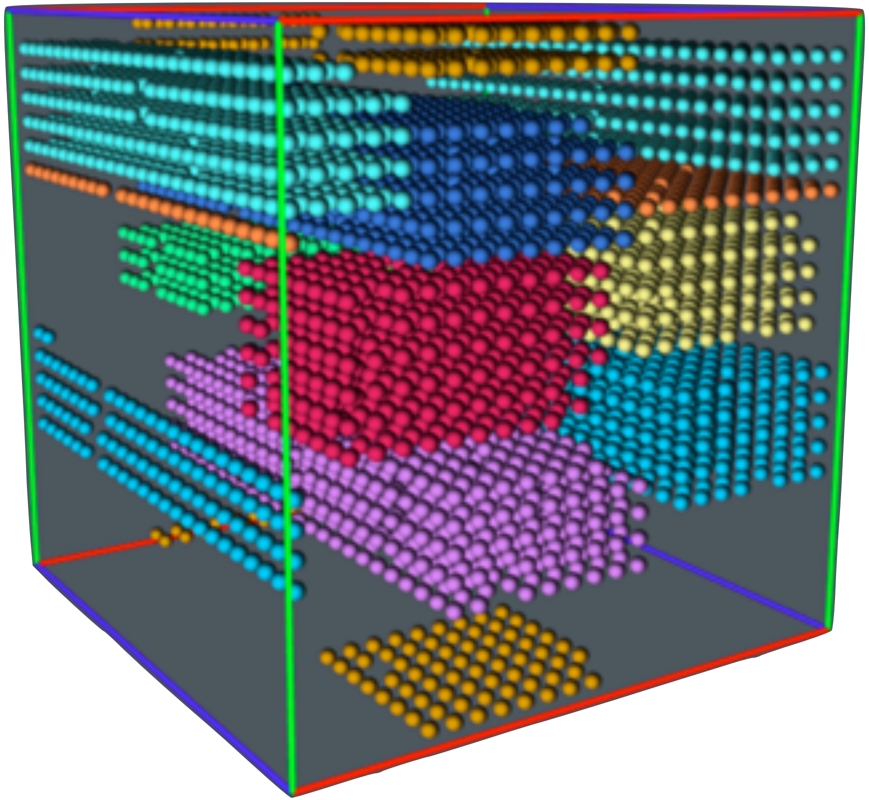
Blue Waters vs Titan



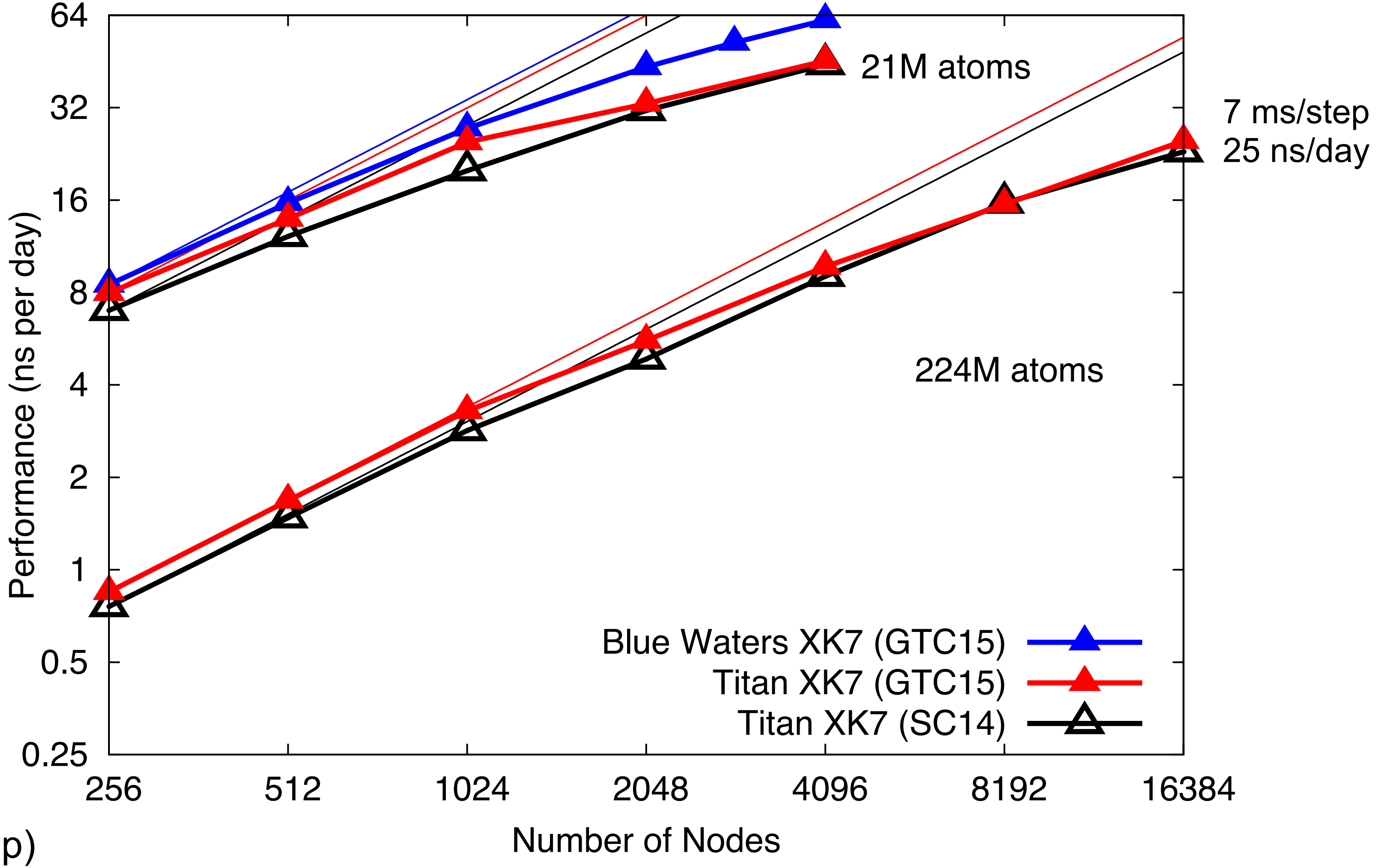
(2fs timestep)

Topology-Aware Scheduling on Blue Waters

- Map jobs to convex sets to avoid network interference
- NCSA, Cray, Adaptive
- Just enabled January 13
- Most likely explanation for Blue Waters performance advantage over Titan
- See Enos *et al.*, CUG 2014

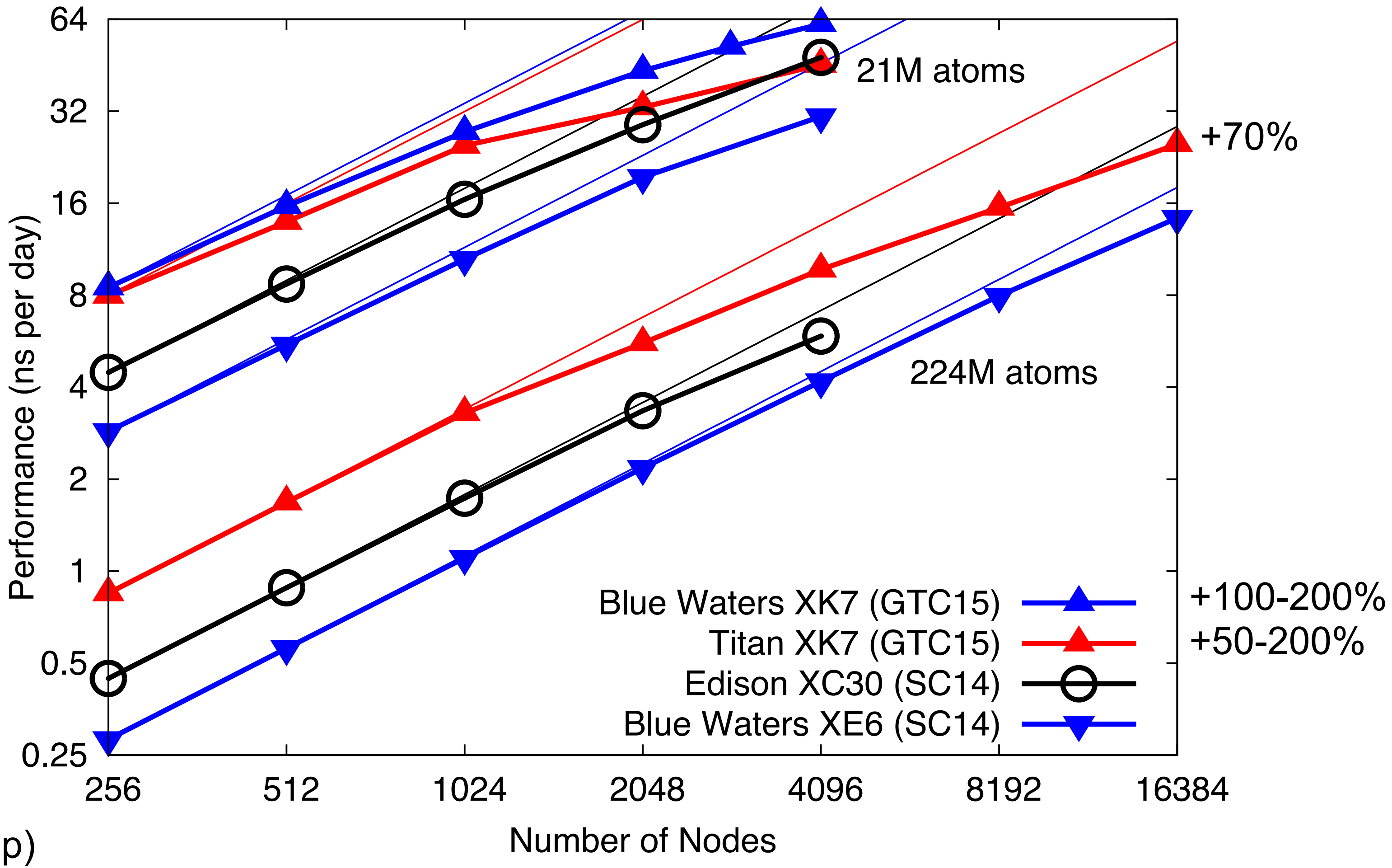


Blue Waters vs Titan



(2fs timestep)

Comparison with CPU-only Machines

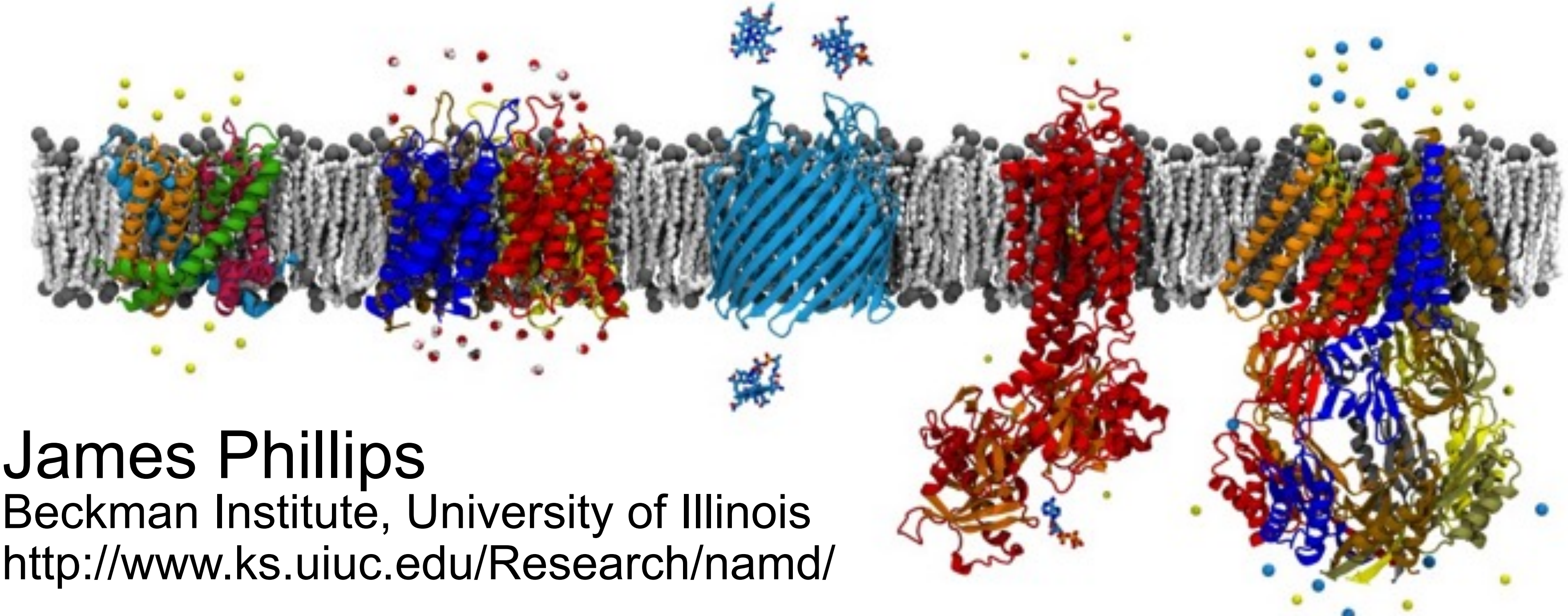


(2fs timestep)

Conclusions

- In biology chemical detail is critical.
- Remote visualization will be necessary.
- Replica exchange enables long timescales.
- Map decomposition to network topology.
- Stream results from GPU in priority order.
- Bad scheduling harms performance.

Thanks to: NIH, NSF, DOE, NCSA,
NVIDIA (**Sarah Tariq**, Patric Zhao, Sky Wu, Justin Luitjens, Nikolai Sakharnykh),
Cray (Sarah Anderson, Ryan Olson), NCSA (Robert Brunner),
PPL (Eric Bohm, Yanhua Sun, Gengbin Zheng, Nikhil Jain)
and 19 years of NAMD and Charm++ developers and users.



James Phillips
Beckman Institute, University of Illinois
http://www.ks.uiuc.edu/Research/namd/